

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Dejan Štepec

**Razvoj postopka za avtomatsko  
ocenjevanje sloga smučarskih skokov**

MAGISTRSKO DELO  
MAGISTRSKI PROGRAM DRUGE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Danijel Skočaj

Ljubljana, 2017



AVTORSKE PRAVICE. Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljane ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

©2017 DEJAN ŠTEPEC





## ZAHVALA

*Zahvaljujem se mentorju izr. prof. dr. Danijelu Skočaju za strokovno pomoč in usmerjanje pri izdelavi magistrskega dela. Hvala tudi družini in vsem prijateljem, ki so me podpirali pri študiju in izdelavi magistrskega dela.*

*Dejan Štepec, 2017*



# Kazalo

**Povzetek**

**Abstract**

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Motivacija . . . . .	1
1.2	Pregled sorodnih del . . . . .	2
1.3	Prispevki . . . . .	5
1.4	Struktura naloge . . . . .	6
<b>2</b>	<b>Nevronske mreže</b>	<b>7</b>
2.1	Klasične nevronske mreže . . . . .	7
2.2	Konvolucijske nevronske mreže . . . . .	16
<b>3</b>	<b>Uporabljene metode</b>	<b>23</b>
3.1	Detekcija smučarskega skakalca . . . . .	23
3.2	Detekcija delov telesa na sliki . . . . .	28
3.3	Detekcija smuči smučarskega skakalca . . . . .	35
3.4	Ocenjevanje sloga smučarskega skoka . . . . .	35
<b>4</b>	<b>Implementacija sistema</b>	<b>43</b>
4.1	Detekcija smučarskega skakalca . . . . .	44
4.2	Detekcija delov telesa in smuči . . . . .	44
4.3	Metoda za ocenjevanje sloga smučarskega skoka . . . . .	46

## KAZALO

<b>5</b>	<b>Podatkovna zbirka smučarskih skokov</b>	<b>51</b>
5.1	Obstoječe zbirke podatkov . . . . .	51
5.2	Podatkovna zbirka smučarskih skokov . . . . .	54
<b>6</b>	<b>Rezultati</b>	<b>59</b>
6.1	Detekcija smučarskega skakalca . . . . .	60
6.2	Detekcija delov telesa in smuči . . . . .	62
6.3	Ocenjevanje sloga smučarskega skoka . . . . .	67
<b>7</b>	<b>Zaključek</b>	<b>75</b>

# Povzetek

**Naslov:** Razvoj postopka za avtomatsko ocenjevanje sloga smučarskih skokov

Smučarski skoki so med Slovenci zelo priljubljen šport, predvsem zaradi uspehov naših športnikov. V magistrskem delu razvijemo metodo za avtomatsko ocenjevanje sloga smučarskih skokov iz videoposnetkov. Kot glavni vir za napoved sodniških ocen uporabimo razporeditev delov telesa in smuči skozi let smučarskega skakalca. Obstoječo metodo za detekcijo delov telesa uporabimo na domeni smučarskih skokov s pomočjo posebne zbirke podatkov, ki jo zgradimo v ta namen. Metodo za detekcijo delov telesa tudi ustrezno spremenimo, da omogoča detekcijo delov smuči. Detektirani deli telesa in smuči tvorijo vhod v metodo za napoved sodniških ocen, kjer uporabimo arhitekturo konvolucijskih nevronske mreže na časovno odvisnih podatkih. Tako naučen model ima napako napovedi, ki je konkurenčna pravim sodnikom.

## Ključne besede

*smučarski skoki, sodniške ocene, konvolucijske nevronske mreže, detekcija delov telesa*



# Abstract

**Title:** Development of an approach for automatic ski jump style scoring

Ski jumping has always been a very popular sport in Slovenia, mostly due to success of our sportsmen. The goal of of this master's thesis is to develop a method for automatic ski jump scoring from videos. As our main source of information we use locations of human body parts along with skis to capture a full body movement of the entire ski jump. We have used an existing method for human pose estimation from images on the domain of ski jumping with the help of specially built dataset. We extend the method for human pose estimation with the support for ski parts detection. Combined locations of human body parts and ski parts represents an input for the method that performs scoring of the ski jump style. The approach is based on convolutional neural networks that are atypically used on a time series data. Our method is able to operate with an error comparable to real judges.

## Keywords

*ski jumping, ski jump scoring, convolutional neural networks, human pose estimation*





# Poglavje 1

## Uvod

### 1.1 Motivacija

Smučarski skoki so med Slovenci zelo priljubljen šport, predvsem zaradi uspehov naših športnikov, ki dosegajo vrhunske rezultate z zmagami v svetovnem pokalu. Veliko vlogo pri smučarskih skokih imajo tudi sodniki, saj je glede na pravila za ocenjevanje skokov [1] le-to sestavljeno iz dolžine skoka in njegovega sloga.

Podobno kot v vsakdanjem življenju se tudi v profesionalnem športu odpirajo priložnosti za uporabo tehnologij, ki bi v šport uvedle višjo stopnjo regularnosti, pri tekmovalcih in ljubiteljih posameznega športa pa večje zaupanje v športne odločitve sodnikov. Sodniške napake so pogost pojav predvsem pri športih, kjer je potek igre zelo hiter in imajo sodniki zelo malo časa za sprejem prave sodniške odločitve. Najočitneje se te težave pojavljajo pri športih z žogo, kjer je posledično v zadnjih letih prišlo do uvedbe različnih tehnoloških rešitev, ki sodnikom olajšajo delo. V nogometu je bil pred svetovnim prvenstvom v Braziliji uveden sistem za avtomatsko detekcijo zadetka [2, 3], ki s pomočjo večih kamer sledi žogi pred golom. Podobni sistemi so bili uvedeni tudi pri tenisu in odbojki, kjer je pomembno detektirati, ali je žoga prečkala črto [4].

Pri smučarskih skokih lahko uporabimo tehnološke rešitve za merjenje

dolžine skoka ter za oceno sloga skoka. Merjenje razdalj je v večini primerov podprto z video analizo [5], vendar še vedno potrebuje operaterja, ki ročno določi mesto pristanka.

Ocenjevanje sloga pa je v večini športov prepuščeno sodnikom, saj gre za kompleksno nalogo z velikokrat slabo definiranimi pravili, ki dopuščajo veliko svobode. V magistrskem delu naslovimo problem ocenjevanja sloga smučarskih skokov na podlagi videoposnetkov. Predlagani sistem avtomatizira ocenjevanje sloga skokov s pomočjo metode za detekcijo delov telesa, ki smo jo prilagodili na specifično domeno smučarskih skokov.

## 1.2 Pregled sorodnih del

Del s področja ocenjevanja sloga smučarskih skokov nismo zasledili, zato se osredotočimo na posamezne podprobleme, ki smo jih reševali, in metode, ki smo jih uporabili.

Najpomembnejši del magistrske naloge predstavlja natančno ocenjevanje pozicije delov telesa in smuči s posameznih slik. To področje računalniškega vida je zelo razvito, kar dokazujejo tudi rezultati na priznanih tekmovanjih [6, 7]. Metode so doživele velik napredek predvsem s pojavom konvolucijskih nevronske mreže, ki so nadomestile uporabo metod z ročno izdelanimi topologijami za interferenco med posameznimi deli telesa in njihovo detekcijo [8, 9, 10]. Začetna dela s področja detekcije delov telesa temeljijo predvsem na modelih, kjer predstavimo posamezne dele telesa in njihovo medsebojno povezanost (Pictorial Structures [11]). Ogrodje je zelo splošno, saj omogoča različne načine opisa delov telesa iz slikovne informacije kot tudi različne načine povezovanja le teh. Za vrednotenje ujemanja modela se nato uporabi različne optimizacijske metode, ki optimizirajo kriterijsko funkcijo, ki računa razdaljo med modelom in detekcijami. Problem takšnih metod je njihova omejena moč, saj lokalni detektorji ne upoštevajo relacij med deli telesa ali jih upoštevajo, zgolj zelo omejeno, saj s topološkim modelom težko opišemo vse možne interakcije med deli telesa. Sposobnost sklepanja o lo-

kaciji dela telesa na podlagi ostalih delov telesa je ključna, saj so določeni deli telesa velikokrat slabo vidni in je vizualne informacije za lokalni detektor premalo.

Metode, ki temeljijo na uporabi konvolucijskih nevronske mreže, detekcijo delov telesa zelo poenostavijo, kajti v primerjavi z grafičnimi modeli ni potrebno izbirati načina opisa predstavitve, topoloških modelov in modelov interakcije med posameznimi deli telesa, saj se tega lahko učinkovito naučimo iz podatkov samih. Ena izmed prvih uspešnih metod je metoda DeepPose [12], ki uporablja konvolucijske nevronske mreže in formulira problem kot regresijo tako, da nauči 7-nivojsko arhitekturo, kjer je vhod celotna slika in izhod vektor koordinat lokacij delov telesa. Avtor nato uporabi kaskado istih arhitektur, le da so te naučene za posamezni del telesa in pričakujejo za vhod okolico detektiranega dela telesa iz prve stopnje. Rezultat so odmiki, ki se uporabijo za premik detekcij, da so te vse bolj točne. Težava pri takšnem pristopu je, da se mora arhitektura naučiti preslikave iz vhodne slike neposredno v koordinate lokacij delov telesa. To je izjemno kompleksna naloga, zato je slabost omenjenega pristopa predvsem natančnost samih detekcij, saj se mora arhitektura naučiti napovedati točno določene vrednosti pozicije dela telesa. Slabost je tudi omejena zmožnost učenja interakcij med deli telesa, saj med učenjem ločeno izboljšujemo lokalne detektorje za posamezne dele telesa, brez dostopa do širšega konteksta, ki ga nudijo ostali detektorji.

Namesto neposredne regresije iz vhodne slike v koordinate detektiranih delov telesa se je uveljavilo kodiranje lokacij delov telesa v verjetnostne mape, ki jih lahko nato v večstopenjskih arhitekturah uporabimo kot vhod poleg vhodne slike in se tako naučimo tudi interakcije med posameznimi deli telesa. Verjetnostno mapo lahko zgradimo tako, da na pozicije delov telesa postavimo 2D Gaussovo porazdelitev in nato računamo kriterijsko funkcijo na verjetnostnih mapah namesto na samem vektorju lokacij delov telesa. Primer takšne metode je metoda CPM [13], ki jo uporabljamo v magistrskem delu. Metoda CPM uporablja večstopenjsko arhitekturo, katere vhod v posamezno stopnjo predstavljajo slikovna informacija ter verjetnostne mape

porazdelitev za posamezne dele telesa iz prejšnje stopnje. Verjetnostne mape omogočajo nadaljnjim stopnjam, da se naučijo slikovnih opisnikov v odvisnosti od modela interakcij med deli telesa, ki ga predstavljajo verjetnostne mape. Vhod v prvo stopnjo metode CPM predstavlja zgolj slika. V delu [14] prvo stopnjo predstavijo kot klasifikacijski problem. Namesto 2D Gaussove porazdelitve uporabijo kar binarno predstavitev ter nato uporabijo znano segmentacijsko arhitekturo VGG-FCN [15], kjer se namesto kriterijske funkcije  $L_2$  uporablja sigmoidna, in sicer glede na posamezni piksel. Sledi regresijski del arhitekture, ki je podoben metodi CPM, le da uporablja zgolj eno stopnjo.

Zgoraj našte metode delujejo na primeru ene osebe. Za posplošitev omenjenih metod na več oseb je potreben detektor oseb. Takšen pristop se je izkazal kot neučinkovit predvsem v primeru prekrivanja več oseb, saj se nam lahko določeni deli telesa pojavljajo večkrat, kar zmoti delovanje omenjenih metod in vodi do napačnih detekcij. To je vodilo do razvoja metod, ki omogočajo detekcijo delov telesa ne glede na število oseb z enovito arhitekturo, ki ni odvisna od ločenih detektorjev oseb [16, 17, 18]. Posebej izpostavimo metodo [18], ki temelji na metodi CPM [13]. Arhitektura je razdeljena na dve veji, pri čemer prva temelji na metodi CPM in je namenjena detekciji delov telesa, druga veja pa zakodira asociacijo med vsemi pari delov telesa v obliki 2D vektorskega polja. To vektorsko polje kasneje uporabijo za grupiranje detektiranih delov telesa po posameznih osebah.

Zanimiva so tudi dela, ki sklepajo o 3D poziciji delov telesa zgolj na podlagi slikovne informacije. Tukaj izpostavimo delo [19], ki prav tako temelji na metodi CPM [13]. Posebnost te metode je neodvisnost med 2D in 3D viri podatkov, kar je ob omejenih velikostih 3D zbirk velika prednost.

K napredku področja računalniškega vida, ki se ukvarja z detekcijo delov telesa, so bistveno pripomogle večje zbirke anotiranih podatkov [6, 7, 20, 21, 22], med katerimi izpostavimo zbirki [6, 7], ki sta zaradi svoje velikosti še posebej pripomogli k razvoju metod, temelječih na uporabi konvolucijskih nevronske mreže.

V drugem delu magistrske naloge uporabljamo rezultate metod za detekcijo delov telesa za potrebe ocenjevanja sloga smučarskih skokov. Dela, ki uporabljajo lokacije delov telesa kot opisnike za klasifikacijo, najdemo na področjih slikovne biometrije, in sicer predvsem pri prepoznavi oseb na podlagi vzorcev hoje [23, 24]. V delu [24] lokacije delov telesa skozi čas v posnetku hoje uporabijo kot vhod v arhitekturo LSTM [25]. Arhitektura LSTM se lahko nauči časovnih odvisnosti med podatki. Podatki o lokacijah delov telesa skozi čas nosijo določene značilnosti o hoji posamezne osebe, na kar kažejo tudi rezultati o uspešnosti prepoznavne oseb omenjene metode. Dela, ki uspešno uporabljajo podatke o poziciji posameznih delov telesa, najdemo tudi na področju prepoznavne akcij. Veliko del prav tako uporablja arhitekture LSTM [26, 27]. Posebej izpostavljamo metodo [28], ki za prepoznavo akcij na podlagi pozicij delov telesa uporablja klasične konvolucijske nevronske mreže. Avtorji zakodirajo surove podatke o lokaciji posameznih delov v umetno sliko, na kateri nato uporabijo enostavno arhitekturo konvolucijskih nevronske mreže. Podoben princip uporabimo tudi v našem delu.

### 1.3 Prispevki

V magistrskem delu razvijemo prototipno rešitev za ocenjevanje sloga smučarskih skokov, ki lahko služi kot dodatni sodnik. Kot glavni vir podatkov uporabimo lokacije delov telesa tekmovalca skozi celoten let. V ta namen raziščemo uporabnost metod za detekcijo delov telesa, ki so se izkazale kot najboljše na različnih lestvicah za testiranje njihove zmogljivosti. Zaradi specifične domene smučarskih skokov in posledično slabega delovanja metod za detekcijo delov telesa predstavimo svojo zbirko anotiranih smučarskih skokov, v kateri smo anotirali skoraj 1800 slik. Izbrano metodo smo nadgradili tudi s podporo za detekcijo smuč. Sistem je sestavljen iz modulov za detekcijo skakalca, detekcijo delov telesa in smuč ter modulom za ovrednotenje in izračun sodniške ocene. Prototipni sistem ovrednotimo na primeru olimpijskih tekem v Vancouvru, kjer se z našim sistemom zelo približamo napaki

obstojećih sodnikov. Celoten sistem temelji na konvolucijskih nevronske mrežah in je zgrajen v modularni zasnovi, ki omogoča enostavno nadgradnjo ali zamenjavo uporabljenih metod.

## 1.4 Struktura naloge

Delo je v grobem razdeljeno na dva dela. V začetnih poglavjih opišemo teoretično ozadje in uporabljene metode, v drugem delu pa opišemo implementacijo in ovrednotenje našega sistema.

V drugem poglavju najprej začnemo s teoretičnimi osnovami, ki so pomembne za razumevanje magistrskega dela. Ker večina dela temelji na uporabi nevronske mreže, pričnemo z opisom klasičnih nevronske mreže. Opis kasneje razširimo z delovanjem konvolucijske nevronske mreže ter opisom njihovih sestavnih delov. V tretjem poglavju opišemo uporabljeno metodo za detekcijo objektov na sliki, ki jo uporabimo za detekcijo smučarskega skakalca. Opišemo metodo za detekcijo delov telesa, ki predstavlja glavni del magistrskega dela in jo uporabimo tudi za detekcijo smuči smučarskega skakalca. Detektirani deli telesa in smuči predstavljajo vhod v metodo za ocenjevanje sloga smučarskega skoka, ki jo ravno tako predstavimo v tem poglavju.

V četrtem poglavju opišemo zgradbo celotnega sistema in implementacijo posameznih metod. V petem poglavju predstavimo glavne zbirke anotiranih podatkov, ki jih potrebujemo za učenje prej omenjenih metod. Predstavimo tudi posebej zgrajeno zbirko anotiranih smučarskih skokov, ki predstavlja prvo tovrstno zbirko podatkov. V šestem poglavju predstavimo rezultate ovrednotenja različnih delov našega sistema.

Delo zaključimo s sklepno diskusijo, kjer povzamemo magistrsko delo in predstavimo možnosti za izboljšave in nadaljnje delo.

## Poglavje 2

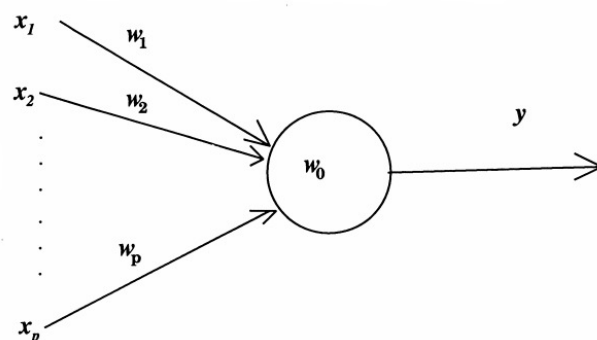
# Nevronske mreže

V tem poglavju predstavimo teoretično ozadje, potrebno za razumevanje magistrskega dela. Ker večina dela temelji na uporabi nevronske mreže, pričnemo z opisom klasičnih nevronske mreže, kar kasneje razširimo z opisom konvolucijskih nevronske mreže. Pri opisovanju si pomagamo predvsem z virom [29], v katerem avtor nazorno opiše delovanje konceptov, ki jih predstavimo v nadaljevanju. Pomagamo si tudi z zapiski in slikovnimi viri predmeta CS231n [30].

## 2.1 Klasične nevronske mreže

### 2.1.1 Perceptron

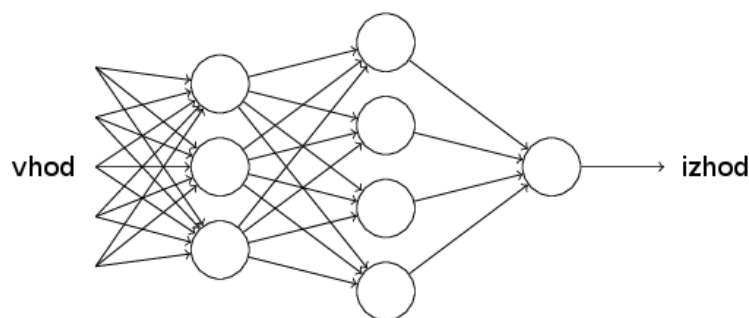
Pri opisu nevronske mreže začnemo z osnovnim gradnikom najbolj enostavnega modela nevronske mreže, perceptronom [31], ki je predstavljen na sliki 2.1. Model perceptrona je bil v osnovi zamišljen kot enostaven matematični model biološkega nevronskega sistema. Perceptron v osnovi deluje tako, da sprejme binarne vhode  $(x_1, x_2, \dots)$ , ki so uteženi z utežmi  $(w_1, w_2, \dots)$ . Izhod perceptrona je prav tako binaren, odvisen od vrednosti utežene vsote in postavljenega praga. Algebraično je model predstavljen z enačbo (2.1).



Slika 2.1: Model perceptrona [29]

$$izhod = \begin{cases} 0 & \text{if } \sum_i w_i x_i \leq \text{prag} \\ 1 & \text{if } \sum_i w_i x_i > \text{prag} \end{cases} \quad (2.1)$$

Na perceptron lahko gledamo kot na odločevalca, ki za svoje odločanje uporablja predstavljena dejstva na vhidih. Z ustrezno nastavitvijo uteži prilagajamo model odločanja glede na to, kaj se nam zdi bolj pomembno. Podobno kot so v biološkem sistemu povezani nevroni, lahko povezujemo tudi perceptrone v kompleksne arhitekture, kot je to predstavljeno na sliki 2.2. Izhode perceptronov prve stopnje povežemo na vhode perceptronov na drugi stopnji ter tako omogočimo modelu, da ne sklepa zgolj na podlagi predstavljenih dejstev, temveč se nauči modelirati odločitve tudi na bolj abstraktnem nivoju.



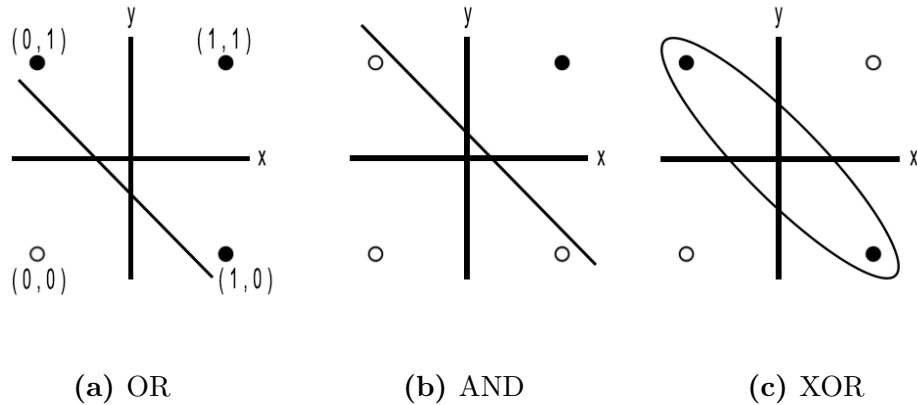
Slika 2.2: Povezovanje perceptronov. Povzeto po [29].



Omejitev enonivojskega perceptrona najhitreje opazimo pri modeliranju logičnih funkcij. Če se osredotočimo na osnovne logične funkcije, ki sprejemajo dva vhoda opazimo, da lahko vhodne podatke ločimo z linearno funkcijo. Opazimo tudi, da perceptron predstavlja linearni model, predstavljen z enačbo (2.2). Prag običajno predstavimo kot svojo spremenljivko, kar modeliramo kot dodatni vhod v perceptron s konstantno vhodno vrednostjo in si tako poenostavimo zapise enačb.

$$x_a w_a + x_b w_b - b = 0 \quad (2.2)$$

Kot prikazuje slika 2.3, sta logični funkciji IN in ALI linearno ločljivi in ju je s perceptronom mogoče zmodelirati z izbiro ustreznih parametrov  $w_a, w_b$  in praga.



**Slika 2.3:** Linearna ločljivost logičnih funkcij

Za računanje optimalnih parametrov perceptrona uporabimo iteracijsko pravilo, ki nam osveži parametre in sčasoma konvergira k optimalnim vrednostim. Naj bo  $D = \{\{x_1, d_1\}, \dots, \{x_s, d_s\}\}$  množica učnih primerov, kjer  $x_i$  predstavlja  $n$ -dimenzionalni vhod in  $d_i$  željeni izhod perceptrona. V primeru logičnih funkcij si lahko  $x_i$  predstavljamo kot 2-dimenzionalni vektor. V splošnem z  $x_{i,j}$  predstavimo  $j$ -to vrednost  $i$ -tega vhodnega primera. Zaradi poenostavitve enačb vpeljemo dodatno dimenzijo vhodnih podatkov  $x_{i,0} = 1$  s konstantno vrednostjo in dodatno utežjo  $w_0$ , ki predstavlja pragovno vre-

dnost. Izhod perceptrona v koraku  $t$  izračunamo z enačbo (2.3), kjer  $f(z)$  predstavlja uporabljeno aktivacijsko funkcijo. V primeru enostavnega perceptrona je to funkcija (2.1) s pragom, premaknjenim na levo stran enačbe. Uteži nato osvežimo z enačbo (2.4). Faktor za stopnjo učenja pri osvežitvi uteži perceptrona ni potreben, saj bi z njim samo skalirali uteži, ne bi pa spremenili predznaka izhoda.

$$y_i(t) = f[w(t) \cdot x_i] = f[w_0(t)x_{i,0} + w_1(t)x_{i,1} + \dots + w_n(t)x_{i,n}] \quad (2.3)$$

$$w_j(t+1) = w_j(t) + (d_i - y_i(t))x_{i,j} \quad 0 \leq j \leq n \quad (2.4)$$

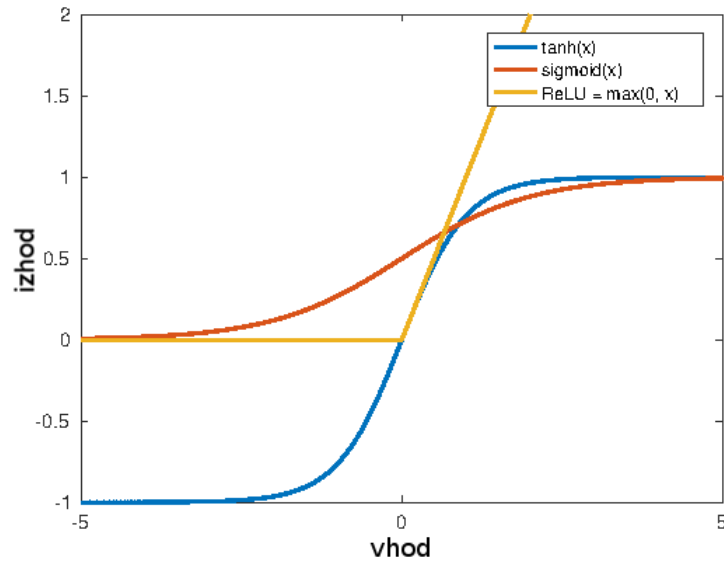
Perceptron bo kot linearni klasifikator konvergirал vedno, kadar bo kot vhod dobil linearno ločljivo množico podatkov. To se jasno pokaže na primeru XOR logične funkcije na sliki 2.3c, kjer vidimo, da z eno linearno funkcijo ne moremo ločiti pozitivnih in negativnih primerov. To sta ugotovila že Minsky in Pappert [32], ki sta prišla do spoznanja, da za rešitev takšnih težav potrebujemo večnivojsko arhitekturo perceptronov, vendar nista našla pravila za učenje takšne arhitekture. Sklepala sta, da takšno pravilo niti ne obstaja, kar se je kasneje izkazalo za napačno. Do odkritja enostavnega pravila za učenje takšnih arhitektur sta minili skoraj dve desetletji. V vmesnem času je bilo v smeri nevronske mreže zelo malo raziskav, saj so se obravnavale kot slepa ulica.

### 2.1.2 Aktivacijske funkcije

Kot smo že omenili, en sam nivo perceptronov ni dovolj za rešitev nelinearnih problemov, kot je modeliranje funkcije XOR. Potrebna je večnivojska arhitektura z vsaj enim skritim nivojem perceptronov. Slabost perceptrona je tudi njegova aktivacijska funkcija, saj lahko z majhno spremembo vrednosti uteži pride do čisto drugačnega rezultata, česar si pri učenju intuitivno ne želimo. Posebej pride težava do izraza ravno pri večnivojskih arhitekturah,

kjer se sprememba propagira skozi več perceptronov. Večnivojske arhitekture zato večinoma uporabljajo sigmoidne nevrone.

Sigmoidni nevron je v osnovi enak perceptronu, spremenjena je zgolj aktivacijska funkcija, tako da majhna sprememba na vходу ali parametrih modela povzroči zgolj majhno spremembo na izhodu. Sigmoidna funkcija je z rdečo predstavljena na sliki 2.4. Izhod sigmoidnega nevrona za  $i$ -ti vhodni primer predstavlja enačba (2.5).



**Slika 2.4:** Aktivacijske funkcije

$$y_i(t) = \frac{1}{1 + \exp(-\sum_j w_j x_{i,j} - b)} \quad (2.5)$$

Zveznost sigmoidne funkcije lahko izkoristimo tako, da spremembo izhoda sigmoidnega nevrona aproksimiramo z linearno funkcijo (2.6).

$$\Delta y_i(t) \approx \sum_j \frac{\partial y_i(t)}{\partial w_j} \Delta w_j + \frac{\partial y_i(t)}{\partial b} \Delta b \quad (2.6)$$

Slabost sigmoidne funkcije je, da je pri velikih in majhnih vrednostih lokalni gradient zelo majhen, kar povzroči, da se sprememba ne propagira. Ta

težava se lahko pojavi tudi pri inicializaciji začetnih parametrov, zato moramo biti previdni, da se ji izognemo. Slabost sigmoidne funkcije je tudi v njenem razponu, saj ni centrirana okoli ničle, kar pomeni, da je vrednost aktivacijske funkcije vedno pozitivna. To se izkaže kot težava predvsem, ko začnemo sigmoidne nevrone povezovati. Če bo vhod v te nevrone vedno pozitiven, bo tudi gradient ves pozitiven ali negativen, kar lahko vpliva na konvergenco. Metodo za učenje večnivojskih nevronske mrež, kjer takšni problemi vplivajo na učenje, predstavimo v delu 2.1.3. V ta namen se uporablja še nekaj drugih aktivacijskih funkcij, namenjenih odpravljanju težav. Hiperbolični tangens, na sliki 2.4 predstavljen z modro, rešuje težavo zaradi centriranja, ohranja pa težavo zaradi nasičenosti nevronov pri velikih ali majhnih vrednostih. S pojavom konvolucijskih nevronske mrež, predstavljenih v nadaljevanju, se je uveljavila aktivacijska funkcija ReLU (Rectified Linear Unit), predstavljena na sliki 2.4 z oranžno barvo. Prednost aktivacijske funkcije ReLU je predvsem v hitrosti konvergence. Hitrost lahko pripišemo linearnemu delu, kjer je gradient konstanten, kar omogoča hitrejšo konvergenco. Enostavnejši je tudi izračun funkcije, saj ne vsebuje eksponentnih delov. Slabost enote ReLU je tako imenovano umiranje nevronov. V primeru, ko je večina vhodov v enoto ReLU negativnih, bo izhod enote enak nič, kar pomeni, da bo tudi gradient ničlen in vhodov ne bo več mogoče spreminjati s postopkom, ki ga opišemo v naslednjem podpoglavju.

### 2.1.3 Postopek vzvratnega razširjanja napake

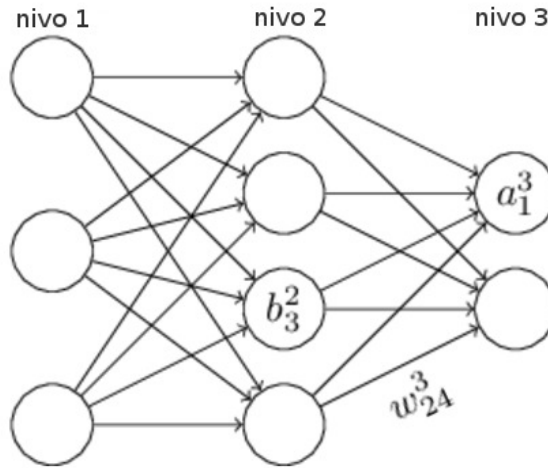
Cilj učenja nevronske mreže je zmanjšati napako, ki jo predstavlja razlika med izhodom nevronske mreže in željenim pravilnim izhodom. Pri pravilu za učenje perceptrona (2.4) smo napako računali kot razliko med izhodom perceptrona ter dejanskim izhodom. V splošnem, kjer izhod ni zgolj binarna vrednost, pa se največkrat uporablja povprečna kvadratna napaka, predsta-

vljena z enačbo (2.7).

$$C(t) = \frac{1}{2s} \sum_i^s ||y_i(t) - d_i||^2 \quad (2.7)$$

Cilj postopka vzratnega razširjanja napake je izračunati parcialne odvode  $\frac{\partial C}{\partial w}$  in  $\frac{\partial C}{\partial b}$  kriterijske funkcije. S pomočjo metode gradientnega spusta nato osvežimo posamezne parametre in tako optimiziramo kriterijsko funkcijo.

Za lažje definiranje postopka najprej definiramo notacijo, ki jo bomo uporabili. Za uteži uporabimo notacijo  $w_{jk}^l$ , s katero predstavimo utež iz  $k$ -tega nevrona v predhodnem nivoju ( $l - 1$  nivo) v  $j$ -ti nevron trenutnega nivoja ( $l$ -ti nivo). Podobno anotacijo uporabimo za aktivacije in pragovne vrednosti. Vizualno je to predstavljeno na sliki 2.5. Aktivacijo  $j$ -tega nevrona v  $l$ -tem nivoju tako lahko zapišemo z enačbo (2.8), kjer  $f$  predstavlja izbrano aktivacijsko funkcijo.



**Slika 2.5:** Notacija zapisa uteži v večnivojski arhitekturi. Povzeto po [29].

$$a_j^l = f\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right) = f(z_j^l) \quad (2.8)$$

Postopek vzvratnega razširjanja napake temelji na štirih enačbah, predstavljenih v obliki, kot jo uporabljajo v delu [29]. Enačba (2.9) predstavlja prvo, ki definira napako na izhodnem nivoju. Enačba (2.10) predstavlja vektorsko obliko enačbe.

$$\delta^L = \frac{\partial C}{\partial a_j^L} f'(z_j^L) \quad (2.9)$$

$$\delta_j^L = \nabla_a C f'(z^L) \quad (2.10)$$

Rekurzivna enačba (2.11) predstavlja napako  $\delta^l$  v odvisnosti od napake na višjem nivoju  $\delta^{l+1}$  in jo zaradi preglednosti zapišemo v vektorski obliki.

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) f'(z^l) \quad (2.11)$$

Če poznamo napako na  $l$ -tem nivoju, se s prvim delom enačbe (2.11) intuitivno pomaknemo nivo višje, tako da napako utežimo z utežmi med nivojema in tako dobimo napako izhoda nivo višje. Z drugim delom enačbe se pomaknemo še skozi aktivacijsko funkcijo, da dobimo napako vhoda. S kombinacijo enačb (2.9) in (2.11) lahko izračunamo napako na vsakem nivoju nevronske mreže.

Enačba (2.12) predstavlja stopnjo spremembe kriterijske funkcije glede na spremembo posamezne pragovne vrednosti.

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \quad (2.12)$$

Enačba (2.13) predstavlja stopnjo spremembe kriterijske funkcije glede na posamezno utež.

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \quad (2.13)$$

Vse štiri enačbe so posledica verižnega pravila pri odvajanju funkcij več spremenljivk.

Algoritem 1 predstavlja metodo za vzvratno razširjanje napake, kjer uporabimo metodo gradientnega spusta za spremembo posameznih parametrov modela. Pri gradientnem spustu uporabimo celotno učno množico. V praksi predvsem zaradi omejene kapacitete pomnilnika uporabimo manjše podmnožice podatkov.

---

**Algoritem 1** Algoritem za vzvratno razširjanje napake

---

```

 $D \leftarrow \{(x_1, y_1), \dots, (x_n, y_n)\}$ 
for all input  $x$  do
     $a^{x,1} \leftarrow x$ 
    // pridobivanje izhoda
    for all layers  $l$  from start+1 do
         $z^{x,l} = w^l a^{x,l-1} + b^l$ 
         $a^{x,l} = f(z^{x,l})$ 
    end for
    // napaka na izhodu
     $\delta^{x,L} = \nabla_a C_x f'(z^{x,L})$ 
    // vzvratno razširjanje napake
    for all layers  $l$  from end-1 do
         $\delta^{x,l} = ((w^{l+1})^T \delta^{x,l+1}) f'(z^{x,l})$ 
    end for
end for
    // osveži uteži in pragovne vrednosti
    //  $\eta$  predstavlja faktor učenja pri gradientnem spustu
     $w^l \leftarrow w^l - \frac{\eta}{n} \sum_x \delta^{x,l} (a^{x,l-1})^T$ 
     $b^l \leftarrow b^l - \frac{\eta}{n} \sum_x \delta^{x,l}$ 

```

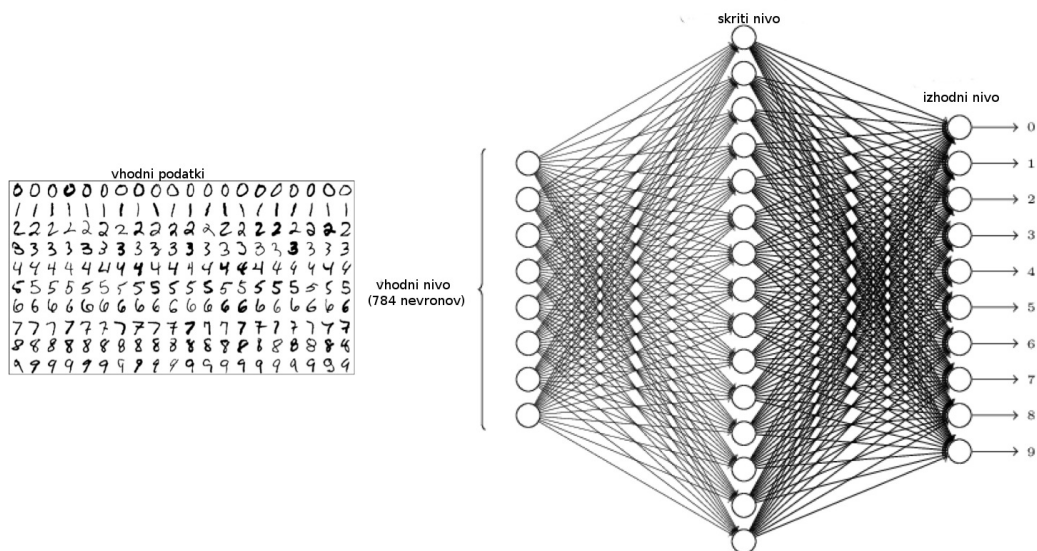
---

## 2.2 Konvolucijske nevronske mreže

Konvolucijske nevronske mreže so zelo podobne klasičnim nevronskim mrežam, le da predpostavljajo, da je vhod slikovna informacija, oziroma povedano bolj splošno, da so vhodni podatki organizirani v večdimenzionalno strukturo.

Kot smo videli v prejšnjem poglavju, so klasične nevronske mreže sestavljene v obliki večnivojske arhitekture. Na vhodni nivo sprejmejo vhodne podatke, ki jih transformirajo skozi serijo skritih nivojev, sestavljenih iz različnega števila nevronov. Posebnost klasičnih nevronskih mrež je, da so nevroni predhodnega nivoja polno povezani z nevroni naslednjega nivoja. Posamezni nevroni so neodvisni in si ne delijo povezav. Na koncu sledi izhodni nivo nevronov, ki predstavljajo izhod nevronske mreže. Težava pri klasičnih nevronskih mrežah je njihovo skaliranje, do česar pride predvsem zaradi njihove polne povezanosti nevronov.

Če se osredotočimo na slikovno informacijo, lahko za primer vzamemo klasifikacijo števil s posameznih slik. Zgled takšnega vhodnega podatka in enostavnega primera nevronske mreže je predstavljen na sliki 2.6.

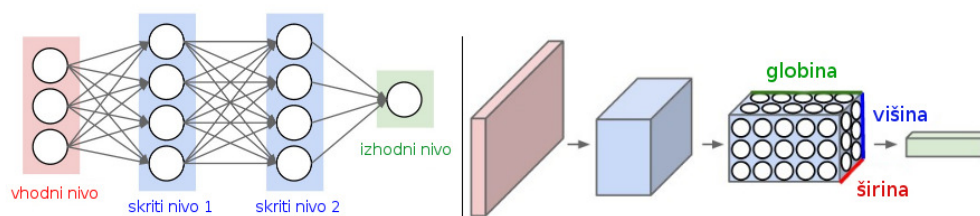


**Slika 2.6:** MNIST [33] klasifikacija števk in enostavna nevronska mreža. Vhodni podatek predstavlja slika posamezne številke. Povzeto po [29].



V primeru zbirke MNIST [33] so vhodne slike velikosti  $28 \times 28$ . Da lahko slikovne vhode uporabimo, jih moramo v primeru klasičnih nevronske mreže transformirati v vektor dolžine 784 ( $28 \times 28$ ). Vsak nevron v skritem nivoju je povezan z vsemi nevroni na vhodnem nivoju, kar znesse 784 povezav na posamezni nevron. Z večanjem števila nevronov v skritem nivoju in večanjem samega števila skritih nivojev se število parametrov hitro povečuje, kar lahko pri učenju vodi do prevelikega prileganja učnim podatkom in posledično do slabega delovanja na testnih podatkih.

Pri konvolucijskih nevronske mrežah so nevroni razporejeni v 3D strukturo tako, da imajo višino, širino in globino. 3D struktura na vходу se pretvori v novo 3D strukturo na izhodu, kot je to prikazano na sliki 2.7. Poleg tega si deli vhodnih podatkov delijo iste nevrone, s čimer se izognemo polni povezanosti, saj si podatki tako delijo uteži (parametre). Intuitivno lahko gledamo na konvolucijske nevronske mreže kot na klasične nevronske mreže, ki jih vzporedno apliciramo na delih vhodnih podatkov. V nadaljevanju opišemo glavne sestavne dele konvolucijskih nevronske mrež.

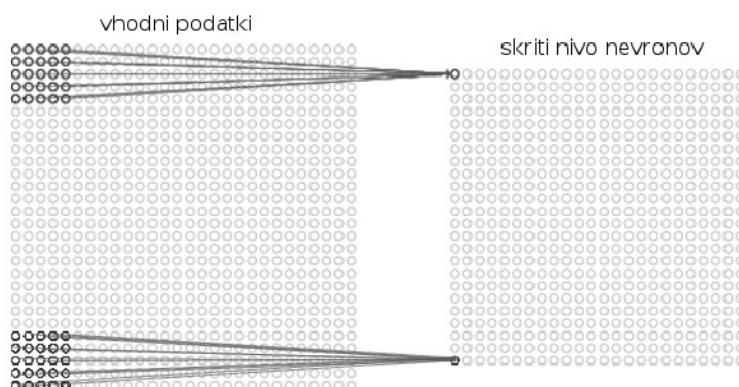


**Slika 2.7:** Vizualizacija arhitekture CNN in 3D struktur. Povzeto po [30].

### 2.2.1 Konvolucijski sloj

Konvolucijski sloj predstavlja glavni gradnik konvolucijskih nevronske mrež. Predstavljamo si ga kot množico 3D filtrov, s katerimi pridobimo aktivacijske mape s pomočjo konvolucije. Posamezni filter je po navadi prostorsko precej manjši, z izjemo globine, kjer zajame celotni vhodni volumen. Uteži filtra, ki ga apliciramo po celotni sliki s pomikajočim se oknom, so parametri, ki

se jih učimo. Rezultat take konvolucije, ki je kar skalarni produkt filtra in območja v vhodnih podatkih, je 2D aktivacijska mapa. Po navadi je posamezni sloj sestavljen iz več takih filtrov in izhodne 2D aktivacijske mape sestavimo v izhodno 3D strukturo, ki jo uporabimo v nadaljnjih slojih. V notaciji klasičnih nevronske mreže si konvolucijske nevronske mreže lahko predstavljamo tako, da je posamezni nevron v skritem nivoju povezan zgolj z majhnim delom vhodnih podatkov (lokalna povezljivost). Vsak skriti nevron pokriva svoje lokalno območje tako, da pokrijemo celotne vhodne podatke. Posamezne skupine nevronov na svojih povezavah uporabljajo enake uteži (deljenje parametrov). Vizualno je to predstavljeno na sliki 2.8.



**Slika 2.8:** Vizualizacija nevronov v konvolucijskih nevronske mrežah. Povzeto po [29].

Prednost konvolucijskega nivoja je ravno v lokalni povezljivosti in deljenju parametrov, s čimer se izognemo prevelikemu številu parametrov, ki bi jih potrebovale klasične nevronske mreže pri polni povezljivosti. Posamezni filter se z istimi parametri aplicira na vseh delih slike. Na filtre lahko gledamo kot na vzorce, ki jih iščemo v vhodnih podatkih. Poleg tega, da se z lokalno povezljivostjo zmanjša število parametrov, je tak način apliciranja filtrov tudi bolj naraven v primeru slikovne informacije, saj se vzorci v slikah lahko pojavijo na različnih mestih v različnem številu. Primeri vzorcev, ki se jih je naučil prvi konvolucijski nivo v arhitekturi AlexNet [34] - ta predstavlja osnovo sodobnih konvolucijskih nevronske mrež -, so predstavljeni na

sliki 2.9. Prednost je tudi v tem, da vhodnih podatkov ni potrebno transformirati ter tako ne pokvarimo strukture podatkov, ki ima pri slikovnih podatkih velik pomen, saj so bližnji slikovni elementi visoko korelirani. V primeru klasičnih nevronske mreže vse vhodne podatke obravnavamo enako, ne glede na njihovo razporeditev, in se mora nevronska mreža sama naučiti strukture vhodnih podatkov.

Pri konvolucijskem sloju imamo več parametrov, s katerimi spreminjamo način apliciranja filtrov na vhodne podatke. Nekateri izmed teh parametrov se uporablja tudi pri ostalih slojih, ki jih opišemo v nadaljevanju. Parametri konvolucijskega sloja so:

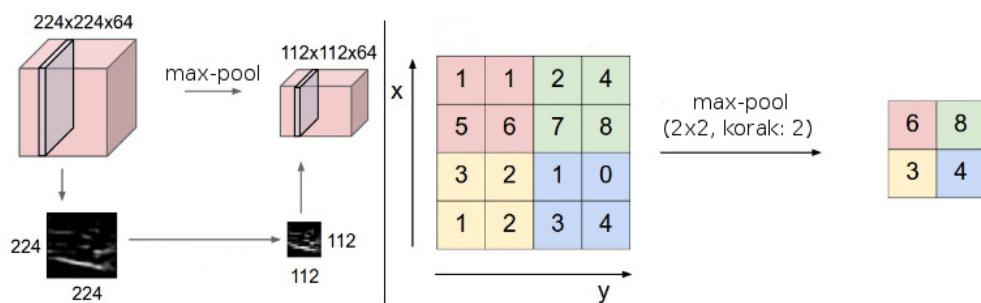
- **število filtrov:** število različnih filtrov, ki jih želimo uporabiti pri konvoluciji;
- **korak konvolucije (ang. stride):** korak, s katerim opravljamo konvolucijo filtra z vhodnimi podatki (npr. korak 1 pomeni, da filter apliciramo na vsakem vhodnem elementu, korak 2 na vsakem drugem ...);
- **razširitev z ničlo (ang. padding):** s tem parametrom nastavljammo način apliciranja filtra na robovih vhodnih podatkov. Če vhodnih podatkov umetno ne razširimo, je izhod iz konvolucijskega sloja drugačnih dimenzij, saj mora biti filter v celoti zajet v vhodnih podatkih.



Slika 2.9: Vizualizacija filtrov v arhitekturi AlexNet [34].

### 2.2.2 Zbirni sloj

Zbirni (ang. pooling) sloj je poseben sloj, ki se je uveljavil v konvolucijskih nevronskih mrežah. Z njim zmanjšamo velikost izhodnih aktivacijskih map, s čimer zmanjšamo število potrebnih parametrov. Podobno kot pri konvolucijskem sloju tudi zbirni sloj apliciramo na lokalnem vzorcu vhodne aktivacijske mape. Vrednosti v lokalnem vzorcu agregiramo glede na uporabljeno agregacijsko metodo. Primer takšne agregacije je, da vzamemo maksimalno ali povprečno vrednost lokalnega vzorca. Vizualno je operacija predstavljena na sliki 2.10, kjer z uporabo filtra velikosti 2 in koraka s stopnjo 2 učinkovito zmanjšamo dimenzijo vhodnih podatkov na četrtnino.

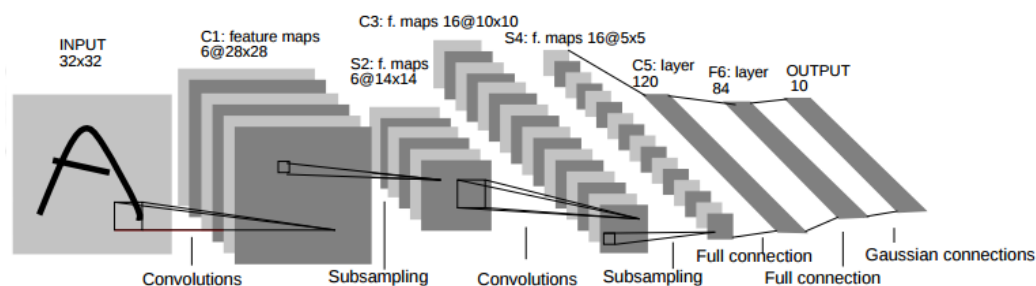


**Slika 2.10:** Vizualizacija delovanja zbirnega sloja. Povzeto po [30].

### 2.2.3 Primer konvolucijske arhitekture

Prvi uspešen primer uporabe konvolucijskih nevronskih mrež najdemo na področju klasifikacije števil. V nadaljevanju predstavimo arhitekturo LeNet-5 [33], ki so jo avtorji uporabili za ta namen in velja za najenostavnejši primer uspešno aplicirane konvolucijske arhitekture. Arhitektura je predstavljena na sliki 2.11.

Vhod v arhitekturo predstavlja normalizirana slika številke, podobno kot pri klasični nevronske mreži na sliki 2.6, le da v tem primeru slike ne vektoriziramo. Vhod je nato povezan s konvolucijskim slojem, sestavljenim iz šestih filtrov velikosti 5 x 5. V prvem sloju se tako arhitektura nauči šest različnih



**Slika 2.11:** Arhitektura LeNet-5 [33], namenjena prepoznavanju števk.

filtru, ki aplicirani na vsako lokacijo v vhodni sliki rezultirajo v šestih izhodnih aktivacijskih mapah. Drugače povedano, arhitektura v vhodnih podatkih išče šest različnih reprezentativnih vzorcev, ki smo se jih naučili v fazi učenja. Kot smo že omenili, bi lahko to operacijo implementirali s klasičnimi nevronskimi mrežami v zaporedni obliki, vendar je implementacija s konvolucijami bistveno učinkovitejša.

Sledi zbirni sloj, kjer zmanjšamo dimenzijo podatkov. Ko s filtri detektiramo posamezne značilke v vhodnih podatkih, ni več pomembno, kje točno se te nahajajo; potrebujemo zgolj njihovo približno lokacijo v odvisnosti glede na ostale. To ni pomembno zgolj zaradi zmanjšanja števila parametrov, saj so vhodni podatki lahko podani v različnih velikostih in se bodo točne pozicije detektiranih značilk spreminjale. Z operacijami, kot sta konvolucija in zbirni sloj, pridobimo na splošnosti naučenih modelov, saj je tako naučen model bolj invarianten na pozicijo in velikost posameznih značilk. Konvolucijski in zbirni sloj se po navadi uporabljata v kombinaciji, saj v višjih slojih po navadi uporabljamo vse več filtrov, ki v kombinaciji z bistveno manjšimi vhodnimi podatki ne rezultira v bistveno večje število parametrov.

Arhitekturo običajno zaključimo s polno povezanim delom, ki deluje popolnoma enako kot pri klasičnih nevronskih mrežah. V polno povezanem delu se nahaja večina parametrov arhitekture. V primeru arhitekture, predstavljene na sliki 2.11, je zadnji sloj sestavljen iz desetih nevronov, kjer posamezni nevron predstavlja števko, ki jo želimo klasificirati.

Sodobne konvolucijske nevronske mreže so dobile nov zagon predvsem po letu 2012 s prepričljivo zmago arhitekture AlexNet [34] na tekmovanju ImageNet [35], kar je predstavljalo nov mejnik razvoja. Danes konvolucijske nevronske mreže prevladujejo skoraj na vseh področjih računalniškega vida, razvitih je ogromno različnih arhitektur in pripadajočih naučenih modelov, ki jih enostavno uporabimo za učenje na svoji problemski domeni. Na razvoj je zelo vplival razvoj strojne opreme, predvsem grafičnih kartic, ki so omogočile razvoj bolj kompleksnih arhitektur. Na razvoj so vplivale tudi večje zbirke podatkov, razne izboljšave in uvedbe novih konceptov.

V nadaljevanju podrobneje predstavimo metodo za detekcijo objektov in metodo za detekcijo delov telesa, ki temeljita na uporabi konvolucijskih nevronskih mrež.

## Poglavje 3

# Uporabljene metode

V tem poglavju predstavimo metodo za detekcijo objektov na sliki, ki jo v magistrskem delu uporabimo za detekcijo smučarskega skakalca. Detekcija smučarskega skakalca v posameznem slikovnem okvirju videposnetka skoka je potrebna za določitev pozicije in velikosti tekmovalca. Predstavimo tudi metodo za detekcijo delov telesa, ki predstavlja glavni del magistrske naloge, saj detektirane dele telesa v nadaljevanju uporabimo kot vhodne podatke v metodo za ocenjevanje sloga smučarskega skoka, ki je ravno tako predstavljena v tem poglavju.

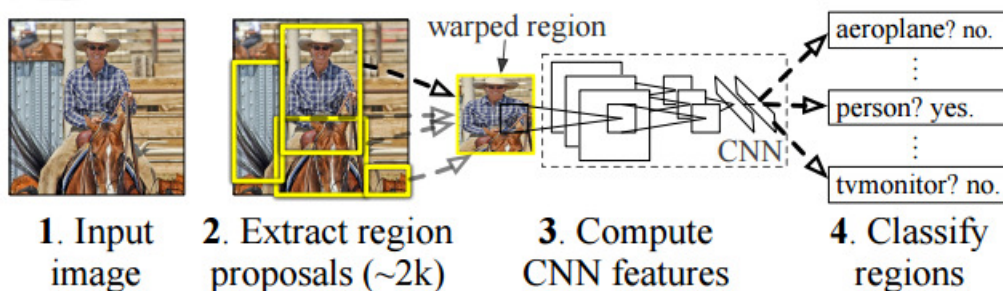
### 3.1 Detekcija smučarskega skakalca

Detekcija objektov na sliki je zelo pomembno raziskovalno vprašanje s področja računalniškega vida. Podobno kot ostala področja je tudi to področje zelo napredovalo predvsem po zaslugi konvolucijskih nevronske mrež. V primerjavi s klasifikacijo, ki smo jo predstavili v prejšnjem poglavju, je detekcija bistveno težja naloga, saj je potreben še postopek lokalizacije objekta na sliki. Lokalizacijo lahko rešujemo na več različnih načinov, najenostavnejša sta premikajoče se okno in generiranje predlogov regij na sliki, ki potencialno vsebujejo objekte, ki jih želimo detektirati. V obeh primerih lahko izkoriščamo obstoječe arhitekture za klasifikacijo objektov.

Če predpostavimo, da imamo rešen problem klasifikacije posameznih slik, lahko k detekciji objektov na sliki pristopimo s pomočjo generiranja predlogov regij na sliki. V prvi fazi generiramo veliko število (nekaj tisoč) predlogov regij v vhodni sliki, ki jih v drugi fazi uporabimo kot vhod v arhitekturo za klasifikacijo. Tako je zgrajen detektor objektov R-CNN [36], ki predstavlja eno prvih implementacij detektorja objektov s pomočjo konvolucijskih nevronske mreže in je na zbirki VOC-2012 [37] dosegla precej boljše rezultate od takrat uveljavljenih metod. V nadaljevanju opišemo inkarnacije metod R-CNN, ki vodijo do metode Faster R-CNN [38], uporabljene v magistrskem delu.

### 3.1.1 R-CNN

Metoda R-CNN [36] je predhodnik metode Faster R-CNN [38] in je sestavljena iz treh delov, ki so predstavljeni na sliki 3.1.



**Slika 3.1:** Zgradba detektorja objektov R-CNN [36].

Prvi del predstavlja metoda za predlaganje regij; v njem so uporabili metodo Selective Search [39], ki v vhodni sliki najde skupine povezanih regij s pomočjo segmentacije in hierarhičnega gručenja v različnih barvnih prostorih. V drugem delu se dobljene regije transformira tako, da so primerne za vhod v arhitekturo AlexNet [34], s pomočjo katere se pridobi opisnike. Tako pridobljeni opisniki v tretjem delu predstavljajo vhod v model SVM, posebej naučen za posamezno kategorijo objektov, ki jih želimo detektirati. S pomočjo metode za odstranjevanje nemaksimalnih regij (ang. non-maximum



suppression) po posameznih kategorijah nato najdemo najboljše detekcije. Za izboljšanje detekcij avtorji uporabijo še regresijski model, s katerim transformirajo detekcije.

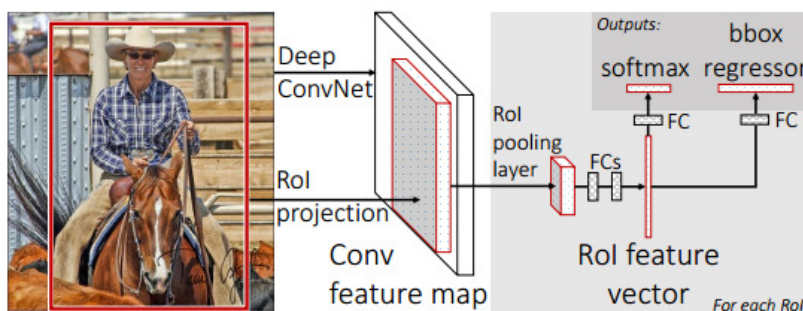
Težava pri omenjenem pristopu je v visoki časovni in prostorski zahtevnosti. Metoda Selective Search [39] se izvaja na CPE in je časovno potratna. Še bolj časovno potratno je apliciranje arhitekture AlexNet [34] na vsaki predlagani regiji v vsaki sliki, saj je predlaganih regij približno 2000 na posamezno sliko. Prostorsko zahtevnost predstavlja predvsem shranjevanje 4096 dimenzionalnih opisnikov, pridobljenih za vsako posamezno regijo. Slabost metode predstavlja tudi njena modularnost, saj ne omogoča enovitega učenja. Izboljšavo predstavlja metoda Fast R-CNN [40], ki jo predstavimo v nadaljevanju.

### 3.1.2 Fast R-CNN

Fast R-CNN [40] predstavlja izboljšavo metode R-CNN [36]. Pri Fast R-CNN apliciramo konvolucijsko mrežo zgolj enkrat na celotni sliki in ne vsakokrat na posamezni predlagani regiji na sliki, saj se regije prekrivajo in večkratno računanje ni učinkovito. To dosežemo tako, da pozicije predlaganih predlogov regij preslikamo v izhodno mapo izbranega konvolucijskega sloja. Avtorji izberejo zadnji konvolucijski sloj in implementirajo poseben sloj, ki nadomesti zadnji zbirni sloj; ta sledi izbranemu konvolucijskemu sloju. Poseben sloj, ki ga avtorji imenujejo sloj *RoI pooling*, deluje zelo podobno kot klasični maksimalni zbirni sloj, le da sloj operacijo aplicira zgolj na delu izhodne mape, kjer je predlagana regija. Operacijo aplicira tako, da je izhod željenih dimenzij, ki ustreza vhodu v nadaljne polno povezane sloje izbrane konvolucijske arhitekture, s čimer na koncu dobimo popolnoma enako arhitekturo kot pri R-CNN. To je pomembno predvsem z vidika ponovne uporabe že naučenih delov iste arhitekture z drugih področij, predvsem s področja klasifikacije slik, s katerimi inicializiramo arhitekturo. Tako inicializirana arhitektura potrebuje za učenje na novi domeni bistveno manj učnih podatkov, ob majhnih zbirkah anotiranih učnih podatkov pa omogoča tudi boljše delovanje. Arhitektura

Fast R-CNN je predstavljena na sliki 3.2.

Fast R-CNN uporablja enovito arhitekturo. Model SVM za klasifikacijo so avtorji nadomestili s slojem Softmax, podobno kot se uporablja pri klasičnem problemu klasifikacije slik s konvolucijskimi mrežami. Prav tako so v isto arhitekturo dodali regresijski del, ki popravi detektirane lokacije objektov. S tako načrtovano arhitekturo lahko učimo vse dele detektorja v enem koraku, kar bistveno pospeši in poenostavi učenje modela. Slabost metode ostaja metoda za predlaganje regij, ki je neodvisna in predstavlja ozko grlo Fast R-CNN metode. Težavo odpravljajo v metodi Faster R-CNN [38], ki jo predstavimo v nadaljevanju.



**Slika 3.2:** Zgradba detektorja objektov Fast R-CNN [40].

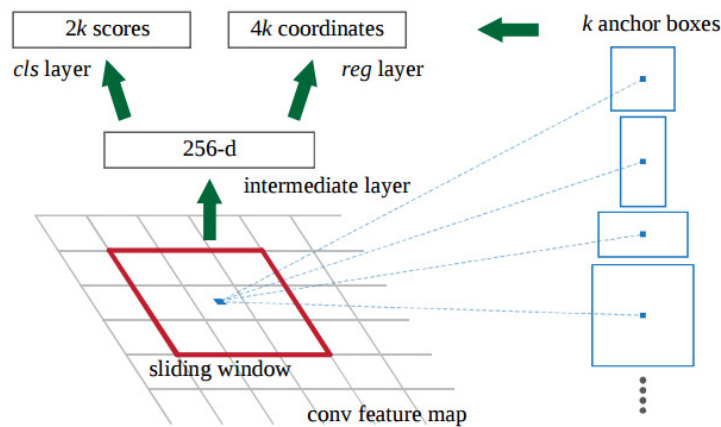
### 3.1.3 Faster R-CNN

Metoda Faster R-CNN [38] nadgradi metodo Fast R-CNN [40] tako, da implementira posebno konvolucijsko arhitekturo, ki je del detektorja objektov Fast R-CNN, namenjena predlaganju regij v vhodni sliki. V primerjavi z metodo Selective Search [39], ki za predlaganje regij porabi približno 2 sekundi, je časovna zahtevnost tako implementirane metode zgolj 10 ms.

Arhitektura RPN (Region Proposal Networks) je zasnovana tako, da si deli del arhitekture z detektorjem objektov Fast R-CNN. Vhod v samostojni del arhitekture RPN predstavlja izhod zadnjega deljenega konvolucijskega sloja, ki se nato uporabi za pridobitev opisnika. Opisnik je pridobljen na

vsaki lokaciji vhoda s pomočjo konvolucije, ki nato predstavlja vhod v regresijski del arhitekture RPN za pridobitev lokacije regije, in verjetnosti, da regija pripada objektu. Intuitivno arhitektura RPN deluje tako, da s pomočjo pridobljenega opisnika na posamezni lokaciji sklepa o poziciji in velikosti potencialnega objekta. Za primer lahko vzamemo, da se je arhitektura naučila opisnika, ki opisuje del avtomobila. Z detektiranim opisnikom na določeni poziciji, ki predstavlja npr. platišče avtomobila, zna arhitektura RPN sklepati o verjetni velikosti in centru avtomobila.

Na vsaki lokaciji arhitektura RPN predlaga  $K$  regij potencialnih objektov. Namesto direktne regresije v pozicijo in velikost posamezne regije so predlagane regije parametrizirane v odvisnosti do  $K$  sidrišč (ang. anchor). Sidrišča so definirana v različnih velikostih in razmerjih stranic. S takim pristopom pridobimo na invariatnosti detektiranih regij. Klasičen pristop s piramidami na vhodnih slikah ali z različnimi velikostmi filtrov bi zahteval večkratno evaluacijo konvolucijske arhitekture. Avtorji uporabijo tri različne velikosti in tri različna razmerja stranic, kar skupno pomeni devet predlaganih regij na posamezni lokaciji. Predlagane regije predstavljajo vhod v del arhitekture za detekcijo objektov, ki ostaja enaka metodi Fast R-CNN. Primer arhitekture RPN in sidrišč je predstavljen na sliki 3.3.



**Slika 3.3:** Primer RPN arhitekture in sidrišč [38].

## 3.2 Detekcija delov telesa na sliki

V tem delu predstavimo uporabljeno metodo za detekcijo delov telesa iz slikovne informacije, to je metodo CPM (Convolutional Pose Machines) [13], ki temelji na metodi PM (Pose Machines) [41], zato najprej opišemo to metodo. Metoda CPM nadgradi metodo PM tako, da idejo pretvori v okvir konvolucijskih nevronske mreže in tako bistveno izboljša delovanje.

### 3.2.1 Pose Machines (PM)

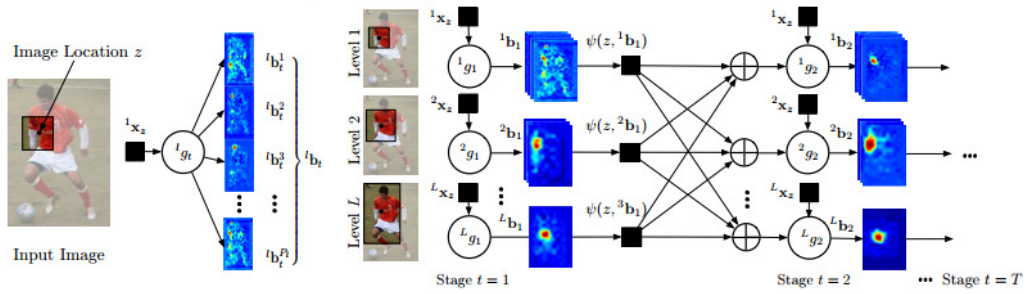
Metoda Pose Machines (PM) [41] rešuje predvsem vprašanje modeliranja interakcije med posameznimi deli telesa. Klasični pristopi uporabljajo metode, ki temeljijo na različnih inkarnacijah drevesnih metod. Pri teh metodah je posebej pomembna zasnova modela za topološki opis razporeditve delov telesa in njihove medsebojne interakcije. Ideja metode PM je, da se modela naučimo samodejno iz samih učnih podatkov in je zasnovana v obliki večstopenjske arhitekture.

Metoda za vsako lokacijo na vhodni sliki in za vsak del telesa vrne verjetnost za prisotnost posameznega dela telesa. Izhod tako predstavlja  $Y = (Y_1, \dots, Y_P)$ , kjer  $P$  predstavlja število delov telesa, in  $Y_p \in Z \subset \mathbb{R}^2$ , kjer je  $Z$  množica vseh  $(x, y)$  lokacij na sliki. Večstopenjska arhitektura je sestavljena iz večrazrednih klasifikatorjev  $g_t(\cdot)$ , ki jih naučimo za detekcijo posameznih delov telesa v posamezni stopnji ( $t$ ) arhitekture. Klasifikator vrne verjetnost  $Y_p = z$ , ki predstavlja verjetnost, da se del telesa  $p$  nahaja na poziciji  $z$ ;  $\forall z \in Z$ . Klasifikator pri tem uporabi opisnike iz vhodne slike  $x_z \in \mathbb{R}^d$  in opisnike iz izhoda klasifikatorja predhodne stopnje, ki predstavlja interakcijo med posameznimi deli telesa. Izhod vsake stopnje je nato vse bolj točna informacija o lokaciji posameznih delov telesa.

Da lahko metoda uspešneje modelira interakcije med posameznimi deli telesa, obenem pa izboljša invariatnost na velikost posameznih delov telesa, se opisnike na posameznih lokacijah  $z \in Z$  na vhodni sliki in izhodih klasifikatorjev izračuna v različno velikih regijah. Na posamezni stopnji arhitekture

imamo tako  $L$  klasifikatorjev, za  $L$  uporabljenih velikosti regij, centriranih na lokaciji  $z$ . Z enačbo (3.1) opišemo delovanje metode, kjer z  ${}^l b_t^p$  opišemo izhod (verjetnost), da klasifikator  ${}^l g_t$  lokaciji  $z$  v vhodni sliki dodeli del telesa  $p$  ob uporabljeni velikosti regije  $l$ . Simbol  $\oplus$  predstavlja združitev opisnikov. Slikovno je arhitektura predstavljena na sliki 3.4. Klasifikatorji prve stopnje arhitekture, ki vrnejo začetno pozicijo predlaganih pozicij delov telesa, sprejmejo kot vhod zgolj slikovne opisnike.

$${}^l g_t \left( x_z^l, \bigoplus_{\substack{l \in 1 \dots L \\ i \in 1 \dots P}} \psi(z, {}^l b_{t-1}^i) \right) \rightarrow \{{}^l b_t^p(Y_p = z)\}_{p \in 1 \dots P} \quad (3.1)$$



**Slika 3.4:** Arhitekture metode Pose Machines [41].

Z  $\psi$  opišemo funkcijo, ki pridobi opisnik iz izhoda klasifikatorja predhodne stopnje. Opisnik izhoda klasifikatorja je namenjen opisu korelacij med posameznimi deli telesa. Opisnik, ki ga uporablja metoda, je sestavljen iz dveh delov. Prvi del je vektorizirana oblika območja, centriranega v točki  $z \in Z$  velikosti  $l \in L$ . Ta opis je namenjen predvsem opisu bližnjih korelacij med deli telesa. Drugi del je namenjen opisu kompleksnejših relacij med oddaljenimi deli telesa. V izhodnih verjetnostnih mapah detektiramo lokacije  $z$  najvišjo vrednostjo in jih opišemo relativno na center  $z$  v polarnem koordinatnem sistemu. Združitev tako dobljenih opisnikov verjetnostnih map in vhodne slikovne informacije predstavlja vhod v posamezne klasifikatorje.

Arhitektura je zasnovana tako, da lahko uporabimo različne klasifikacijske metode ter različne slikovne opisnike. Avtorji za klasifikacijo uporabijo

metodo naključnih gozdov (ang. Random Forests) [42], za slikovne opisnike pa opisnike HOG [43].

### 3.2.2 Convolutional Pose Machines (CPM)

Metoda CPM [13] nadgradi metodo PM [41] tako, da za arhitekturo uporabi konvolucijske nevronske mreže. Tako implementirana metoda ne potrebuje ročno definiranih opisnikov, kot jih uporablja metoda PM, saj se optimalnih opisnikov arhitektura nauči iz podatkov.

Podobno kot metoda PM je tudi metoda CPM zgrajena v obliki več-stopenjske arhitekture. Primer dvostopenjske arhitekture je predstavljen na sliki 3.5. Podobno kot pri metodi PM je vhod v prvo stopnjo zgolj slikovna informacija. Prva stopnja je na sliki 3.5 označena s črnim okvirjem. Vhod predstavlja normalizirana slikovna informacija velikosti  $368 \times 368$ . Prva stopnja arhitekture je sestavljena iz petih konvolucijskih slojev, katerim sledita dve  $1 \times 1$  konvoluciji. Vloga  $1 \times 1$  konvolucij je zmanjšanje/povečanje dimenzionalnosti podatkov. Vsakega izmed končnih ( $P$ )  $1 \times 1$  filtrov si lahko predstavljamo kot klasifikator posameznega dela telesa ( $P$  predstavlja število vseh delov telesa), ki za posamezno 2D lokacijo 3D izhoda predhodnega konvolucijskega sloja (iz podatkov naučen  $\mathbb{R}^d$  opisnik za posamezno lokacijo) določi verjetnost pojavitve dela telesa.

PM metoda uporablja na posamezni lokaciji  $z \in Z$  regije različnih velikosti, da doseže večjo invariatnost na velikost delov telesa na slikah, v stopnjah  $t > 1$  pa tudi zaradi zmožnosti modeliranja interakcij med posameznimi deli telesa. Pri CPM doseženo to implicitno s kombinacijo konvolucijskih in zbirnih slojev. Tabela 3.1 predstavlja efektivno velikost regij posameznih konvolucijskih slojev prve stopnje relativno na vhodno sliko.

Vhod v stopnjo  $t > 1$  podobno kot pri metodi PM predstavlja izhod iz predhodne stopnje in slikovne informacije. Vhod slikovne informacije na posamezno stopnjo je na sliki 3.5 označen z modro barvo. Modri del predstavlja del arhitekture, ki je enak vsem stopnjam. Arhitektura je popolnoma enaka delu prve stopnje. Četrty konvolucijski sloj se v vsaki stopnji uporabi neodvi-

**Tabela 3.1:** Velikost regij vhodne slike po posameznih konvolucijskih slojih prve stopnje CPM metode.

konvolucijski sloj	velikost
1	9 x 9
2	26 x 26
3	60 x 60
4	96 x 96
5	160 x 160

sno, tako damo modelu možnost prilagajanja slikovnega opisnika posamezni stopnji. Vhod v drugo stopnjo na sliki 3.5 tako predstavlja opisnik slikovne informacije, rezultati predhodne stopnje (lokacije delov telesa, zakodirane v obliki verjetnostnih map) in lokacija centra osebe, katere dele telesa želimo detektirati. Na vhodnih slikah se lahko pojavi več oseb, ki so si blizu skupaj, in s podatkom o centru osebe poizkušamo osredotočiti arhitekturo na željeno osebo.

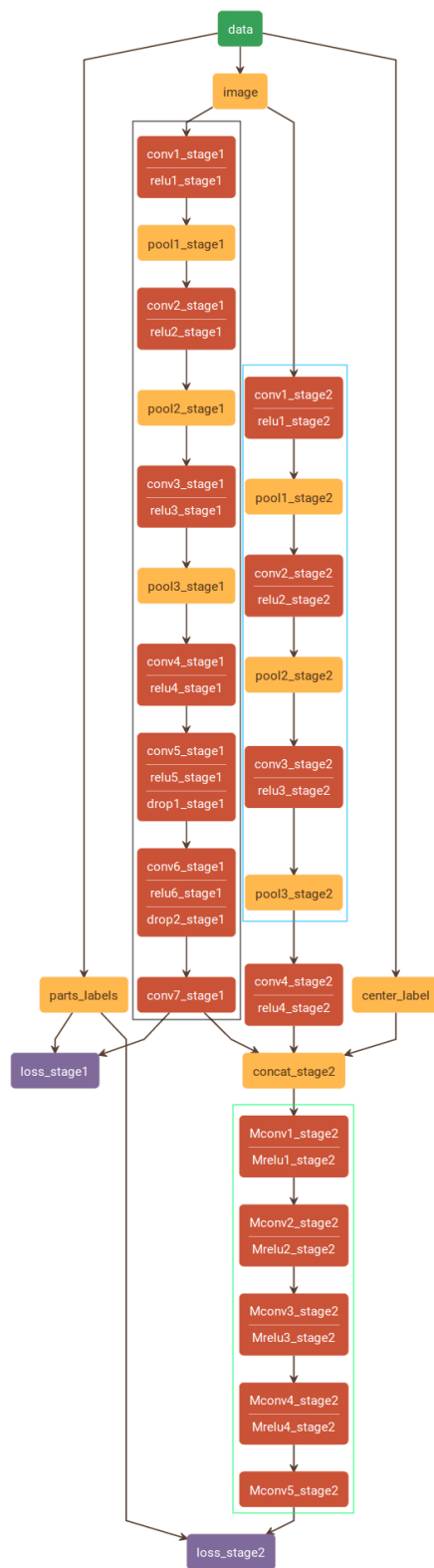
Vsaka stopnja arhitekture  $t > 1$  nato tako združen vhod podatkov sprocesa z arhitekturo, na sliki 3.5 označeno z zeleno barvo. Za razliko od metode PM, ki definira posebne opisnike tudi za verjetnostne mape izhoda predhodne stopnje, se CPM del arhitekture, označen z zeleno barvo, sam nauči, katere dele vhodnih podatkov bo uporabil ter tudi, kako jih uporabiti. Definirani so zgolj parametri konvolucijskih in zbirnih slojev ter njihovo število. Ti parametri definirajo efektivno velikost regije okoli posamezne pozicije v vhodnih podatkih, do katere ima model dostop.

Tako kot je zasnovana arhitektura, lahko vsebuje zelo veliko slojev. Pri učenju tako velike arhitekture lahko pride do težav zaradi pojemanja gradienta, saj se ob vzvratnem razširjanju napake le-ta množi z majhnimi vrednostmi uteži, kar hitro vodi do praktično ničelnega gradienta in posledično do nezmožnosti učenja arhitekture. Metoda CPM težavo odpravlja tako, da kriterijsko funkcijo računa na koncu vsake stopnje in tako ohranja dovolj visoko magnitudo gradienta za uspešno učenje vseh slojev arhitekture. Kri-

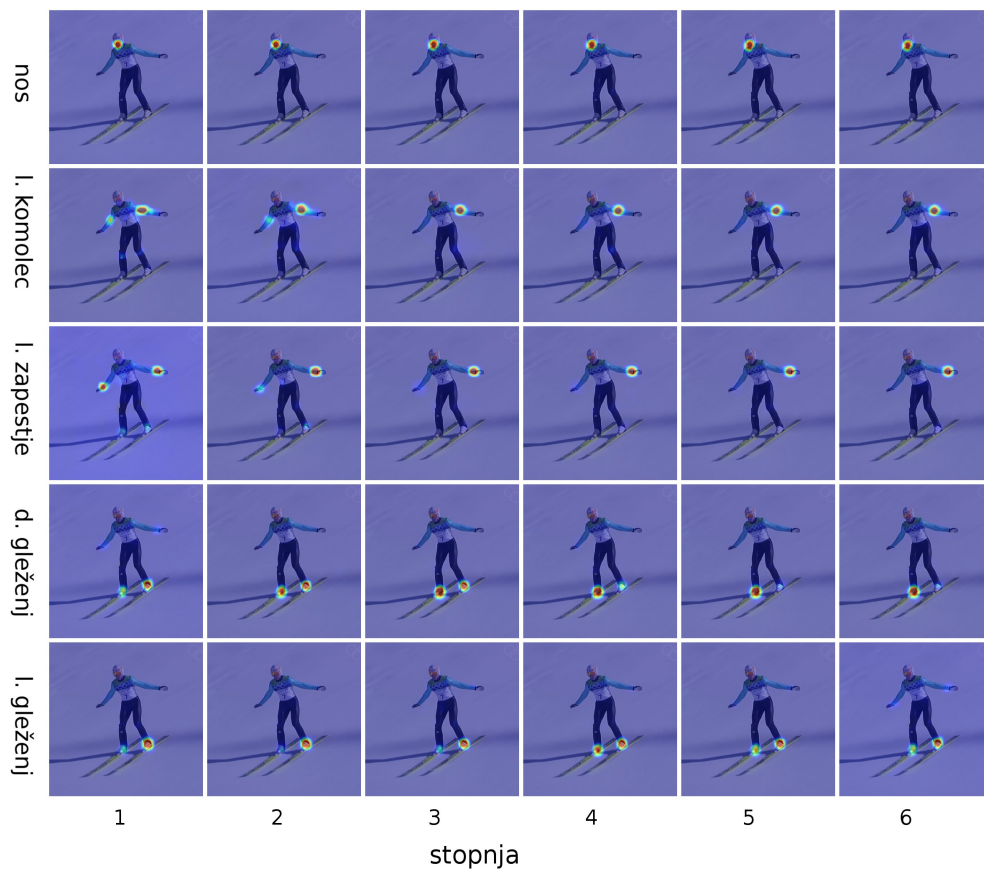
terijska funkcija predstavlja  $L_2$  razdaljo med idealno verjetnostno mapo in mapo, ki jo vrne vsaka posamezna stopnja arhitekture. Idealno verjetnostno mapo zgradimo tako, da na prave pozicije delov telesa postavimo 2D Gussove funkcije z majhno varianco. Posebnost prve stopnje je, da anotiramo vse osebe v vhodni sliki in ne zgolj osebe, za katero skušamo detektirati dele telesa. Razlog je v tem, da prva stopnja še ne pozna konteksta interakcij med posameznimi deli telesa in bi neanotirani deli telesa negativno vplivali na učenje, saj bi za morebitne iste dele telesa obstajali anotirani kot tudi neanotirani podatki. V primeru smučarskih skokov je vedno prisotna zgolj ena oseba, tako da način implementacije ni pomemben.

Slika 3.6 prikazuje delovanje metode CPM. Na izhodu prve stopnje opazimo, da so simetrični deli telesa še slabo ločeni, kar je logično, saj metoda v prvi stopnji še ne pozna topološke ureditve delov telesa in o lokaciji sklepa zgolj na podlagi vizualne informacije s slike. Izjema je nos, ki je dobro detektiran že v prvi stopnji, saj lahko o njegovi lokaciji dokaj nedvoumno sklepamo že zgolj samo iz slikovne informacije. Opazimo, da se lokacije skozi več stopenj izboljšujejo, kar pomeni, da se arhitektura implicitno nauči topološkega modela in modela interakcij človeškega telesa. Še posebej je to vidno v primeru desnega gležnja, ki je na prvi stopnji detektiran na poziciji levega gležnja in se skozi nekaj stopenj lokacija popravi. Vzrok leži v izboljšanju lokacij ostalih delov telesa in ko v višjih stopnjah dobimo kot vhod tudi te podatke, zna arhitektura o lokaciji gležnja sklepati na podlagi lokacij ostalih delov telesa. Arhitektura ustrezno popravi lokacijo, da detekcija ustreza implicitno naučenemu modelu telesa.





**Slika 3.5:** Implementacija metode CPM [13] s konvolucijskimi nevronske mrežami.



**Slika 3.6:** Verjetnostne mape za posamezne dele telesa (nos, levi komolec, levo zapestje, desni gleženj, levi gleženj) na izhodih posamezne stopnje (6 stopenj) metode CPM [13].

### 3.3 Detekcija smuči smučarskega skakalca

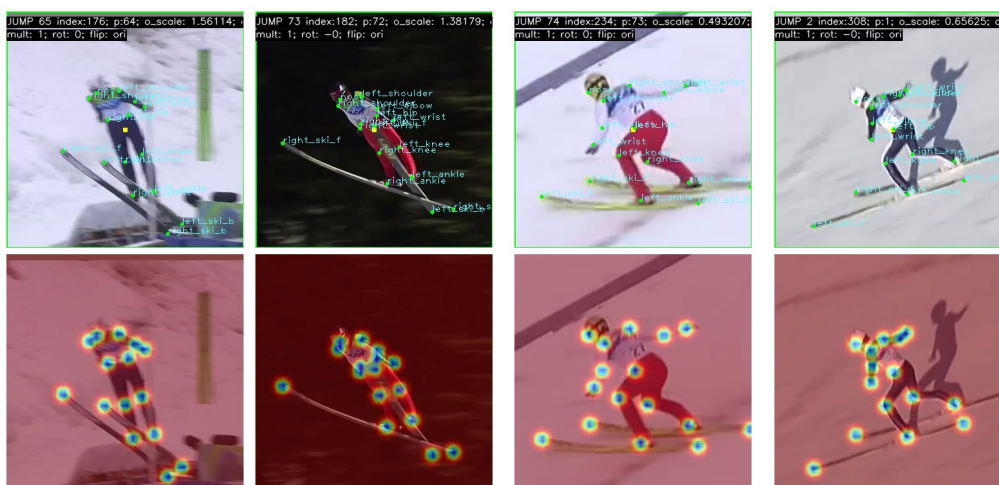
Za detekcijo smuči smučarskega skakalca uporabimo metodo CPM [13], tako da smuči obravnavamo kot dodatni del telesa. Let smučarskega skakalca, ki vpliva na sodniško oceno, lahko glede na razporeditev delov telesa in smuči razdelimo v štiri faze. Prvo fazo predstavlja odskok, drugo fazo predstavlja let skozi hrbtnište skakalnice, tretjo doskok in četrto vožnja po doskoku (iztek). Opazimo, da se pozicija smuči razlikuje glede na fazo leta, obenem pa je njihova pozicija odvisna tudi od ostalih delov telesa.

Podobno kot se metoda CPM nauči topološkega modela razporeditve delov telesa in modela njihove interakcije, se lahko naučimo modelirati pozicijo smuči, saj je ta odvisna od pozicije ostalih delov telesa. Smuči opišemo z njihovimi skrajnimi točkami ter tako dobimo dodatne štiri anotacije lokacij delov telesa v obliki, ki jo lahko uporabimo v metodah za detekcijo delov telesa. Na sliki 3.7 so predstavljene faze leta in vhodni podatki v CPM metodo. Prva vrstica predstavlja anotirane podatke iz podatkovne zbirke smučarskih skokov, druga vrstica pa vhod v učenje metode CPM, kjer smo na posamezne lokacije delov telesa postavili 2D Gaussove funkcije z majhno varianco.

### 3.4 Ocenjevanje sloga smučarskega skoka

Slog smučarskega skoka ocenjuje pet sodnikov, pri čemer posamezni sodnik lahko za skok dodeli maksimalno 20 točk. Najboljša in najslabša ocena se ne upoštevata in tako lahko tekmovalec dobi maksimalno 60 točk. Pravila za ocenjevanje smučarskih skokov so zapisana v pravilniku mednarodne smučarske zveze [1].

Sodniki ocenjujejo skok od odskoka s skakalnice do vožnje mimo črte, ki označuje konec območja ocenjevanja. Pravila za ocenjevanje so razdeljena v tri faze leta (let, doskok, iztek), za katere so pravila ocenjevanja predstavljena v nadaljevanju. Pravila so povzeta po pravilniku [1].



(a) odskok

(b) let

(c) doskok

(d) iztek

**Slika 3.7:** Primer faz leta in primerov anotacij vhodnih podatkov smuči in ostalih delov telesa, ki predstavlja vhod v metodo CPM [13].

### 3.4.1 Pravila za ocenjevanje sloga smučarskega skoka

#### Ocenjevanje odskoka in leta skakalca

Skakalec mora narediti agresiven odskok, iz katerega nato tekoče preklopi v fazo leta in pravočasno začne priprave na doskok. Sodniki ocenjujejo, kako dobro skakalec izkorišča aerodinamične lastnosti, kar skakalec doseže z ustrezno pozicijo delov telesa. Leva in desna polovica telesa morata biti simetrično poravnani s stegnjenimi nogami. Sodniki lahko za to fazo leta odbijejo največ pet točk.

#### Pristanek skakalca

Skakalec mora v fazo doskoka preiti iz stabilne pozicije faze letenja. V fazo doskoka skakalec preide tako, da dvigne glavo in zgornji del telesa, začne z dvigovanjem rok in postavljanjem smuči v vzporedno pozicijo. Tik pred pristankom mora skrčiti kolena za uspešno absorbcijo energije pri doskoku. Pristanek mora biti tekoč in mehek, kar skakalec doseže z uporabo moči in prave razporeditve delov telesa. Skakalec poveča razdaljo med nogama in

potisne eno izmed nog naprej, kar vodi v pozicijo telemarka. Roke mora iztegniti pravokotno na smuči, in sicer takoj po pristanku.

Sodniki ocenjujejo prehod iz faze letenja v fazo doskoka, predvsem poravnost zgornjega dela telesa, pozicijo nog in skrčenost kolen, s čimer skakalec absorbira energijo ob pristanku. Skrčenost kolen ne sme biti prevelika. Skakalec mora doskočiti v izpadni korak (telemark), ki se ocenjuje posebej. Pri izpadnem koraku mora biti razdalja med konico zadnje noge in peto prve vsaj polovico dolžine stopala. Razmak med smučmi naj ne bi bil večji od dveh širin posamezne smuči. Sodniki lahko v tej fazi leta odbijejo največ pet točk. Dodatni dve točki lahko odbijejo za nepravilno izvedbo telemarka.

### **Vožnja skakalca v izteku**

Po pristanku mora skakalec obdržati pozicijo telemarka še približno 10 - 15 metrov. Do črte, ki označuje konec ocenjevanja, mora skakalec obdržati višjo stabilno držo s poravnanimi smučmi ali v ustrezni trikotni plužni poziciji, s katero zmanjša hitrost.

Sodniki ocenjuje ohranitev telemark pozicije (10 - 15 metrov), pozicijo smuči (vzporedna ali plužna) in enakomerno uravnoteženost skakalca na obeh smučeh s stegnjenim zgornjim delom telesa. Tekmovalec mora stabilno držo obdržati do črte, ki označuje konec ocenjevanja. Za nestabilno ali nepravilno pozicijo delov telesa lahko sodniki odbijejo od 0.5 do 3 točke. Če se skakalec ob doskoku z delom telesa dotakne smuči ali skakalnice, dobi odbitek od 4 do 5 točk. Če skakalec ob doskoku pade, dobi odbitek sedmih točk, kar je tudi maksimalni odbitek pri ocenjevanju vožnje v izteku.

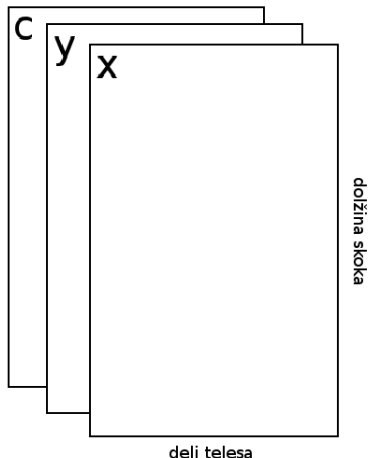
### **3.4.2 Metoda za ocenjevanje sloga smučarskega skoka**

Iz zgoraj opisanih pravil za ocenjevanje smučarskega skoka opazimo, da so ta definirana zelo ohlapno in subjektivno. Pravil v taki obliki ni mogoče neposredno uporabiti, saj jih ne moremo opisati matematično. Izjema je zgolj izvedba telemarka, kjer so pravila definirana nekoliko bolj natančno. Namesto opisovanja pravil v algoritmični obliki v magistrskem delu rešujemo

težavo tako, da razvijemo metodo, ki se pravil nauči sama na osnovi ocen pravih sodnikov.

Za pridobivanje ocene sloga skoka uporabimo podatke o poziciji delov telesa in smuči skozi let posameznega skakalca. Podobno kot se metoda za detekcijo delov telesa implicitno nauči topološkega modela in modela interakcij med posameznimi deli telesa, je cilj naše metode, da se nauči modela za ocenjevanje sloga smučarskega skoka. Glede na opisana pravila za ocenjevanje ima pozicija delov telesa ključno vlogo in bi morali nositi vso potrebno informacijo za uspešno ocenjevanje.

Pri tako zbranih podatkih imamo opravlja tudi s časovno dimenzijo, saj je pozicija delov telesa v času  $t$  odvisna od pozicije v času  $t'$ , kjer je  $t' < t$ . Odločimo se, da podatke predstavimo v obliki slikovne informacije, kot je to predstavljeno na sliki 3.8. Dejanski vhod v metodo je predstavljen na sliki 3.9.



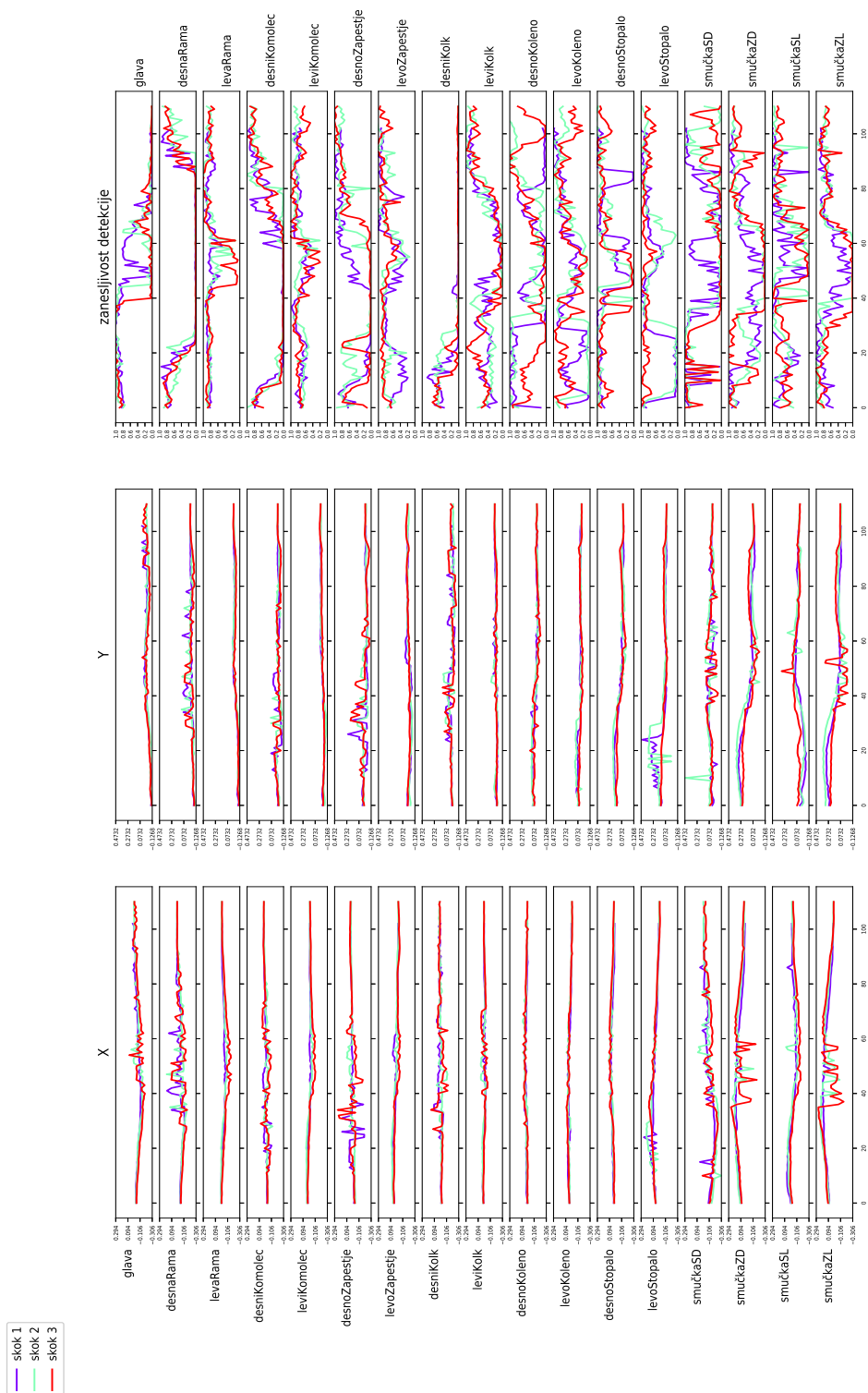
**Slika 3.8:** Predstavitev vhodnih podatkov v metodo za ocenjevanje smučarskega skoka. Vhod je organiziran v 3D slikovno informacijo, kjer prvi kanal predstavlja koordinate  $x$  lokacij delov telesa, drugi kanal koordinate  $y$  in tretji kanal zanesljivost detekcije. Posamezna vrstica predstavlja sliko smučarskega skoka, dolžina vrstice je enaka številu uporabljenih delov telesa. Koordinate normaliziramo glede na masno središče tekmovalca.

Za predstavitev v obliki slikovne informacije smo se odločili zato, da lahko uporabimo konvolucijske nevronske mreže, ki lahko tako izkoristijo tudi časovno dimenzijo. Podoben princip so uporabili tudi v delu [28], kjer so tako obliko vhodnih podatkov uporabili za klasifikacijo akcij v videoposnetkih s pomočjo konvolucijskih nevronskih mrež.

Problem zastavimo v obliki regresije, tako da napovedujemo ocene petih sodnikov. Uporabimo konvolucijsko arhitekturo LeNet-5 [33], ki jo ustrezno spremenimo za reševanje regresijskega problema. V ta namen uporabimo kriterijsko funkcijo  $L_2$  in ustrezno spremenimo dimenzijo izhoda zadnjega polno povezanega sloja, da ustreza številu sodnikov. Kriterijska funkcija je predstavljena z enačbo (3.2).

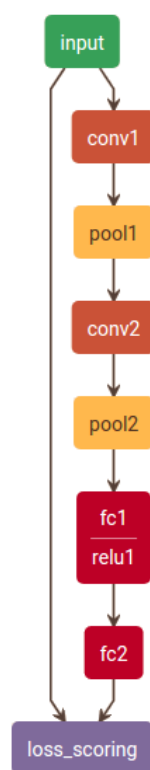
$$loss\_scoring = \frac{1}{2N} \sum_{n=1}^N \|\hat{y}_n - y_n\|_2^2 \quad (3.2)$$

Z  $\hat{y} = [y_1, \dots, y_5]$  predstavimo referenčne sodniške ocene za posamezen skok. To so ocene, ki so jih pravi sodniki dodelili posameznemu skoku. Z  $y$  predstavimo sodniške ocene, ki jih vrne predlagana metoda. Arhitektura je predstavljena na sliki 3.10. Za to arhitekturo smo se se odločili zaradi njene enostavne zgradbe in majhnega števila parametrov, saj smo bili omejeni z velikostjo podatkovne zbirke. Anotirano zbirko podatkov smučarskih skokov predstavimo v nadaljevanju.



**Slika 3.9:** Vizualizacija vhoda v metodo za ocenjevanje sloga smučarskega skoka. Vizualizacija prikazuje tri smučarske skoke, kjer so detekcije normalizirane glede na detektirano masno središče tekmovalca.





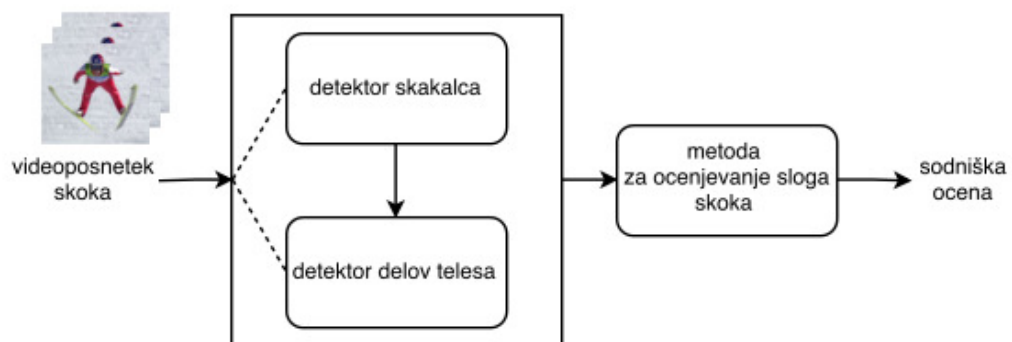
**Slika 3.10:** Zgradba arhitekture, ki smo jo uporabili.



## Poglavje 4

# Implementacija sistema

V tem poglavju predstavimo implementacijo sistema za avtomatsko ocenjevanje sloga smučarskih skokov iz videoposnetka. Sistem je sestavljen iz metod, ki smo jih predstavili v prejšnjem poglavju, in je zasnovan v modularni zasnovi, ki omogoča enostavno menjavo posameznih metod. Na sliki 4.1 je predstavljena zgradba sistema. Celoten sistem je zasnovan na metodah, ki temeljijo na uporabi konvolucijskih nevronske mreže. Sistem je implementiran v programskem jeziku Python. Za delo s konvolucijskimi nevronske mreže uporabljamo programsko ogrodje Caffe [44]. V nadaljevanju podrobneje opišemo implementacijo posameznih delov sistema.



**Slika 4.1:** Zgradba sistema za avtomatsko ocenjevanje smučarskega skoka.

## 4.1 Detekcija smučarskega skakalca

Detekcijo smučarskega skakalca smo implementirali z detektorjem objektov Faster R-CNN [38]. Uporabili smo javno dostopno programsko kodo, ki smo ji dodali podporo za našo podatkovno zbirko smučarskih skokov. Arhitekturo smo inicializirali s pomočjo že obstoječega modela za detekcijo objektov, naučenega na podatkovni zbirki COCO [7]. Popravili smo le število razredov v polno povezanem sloju, saj v našem primeru detektiramo zgolj skakalca. Arhitekturo smo nato naučili na naši podatkovni zbirki. Naučeni model nato uporabimo v modulu za detekcijo skakalca, ki kot vhod sprejme posamezno sliko, kjer detektira skakalca. Detekcija skakalca je pomembna predvsem za delovanje detektorja delov telesa, saj smo tako dobili lokacijo in približno velikost skakalca. Tako uporabimo detektor delov telesa zgolj pri eni vhodni velikosti, kar poveča hitrost delovanja. Za še hitrejše delovanje bi lahko uporabili tudi sledilnik ali pa kakšno hitrejšo metodo za detekcijo objektov, kot je metoda SSD [45].

## 4.2 Detekcija delov telesa in smuči

Detekcija delov telesa in smuči predstavlja glavni del magistrske naloge. Uporabili smo metodo CPM [13]. Sprva smo poizkusili uporabiti že naučeni model za detekcijo delov telesa. Ugotovili smo, da je pozicija delov telesa pri smučarskih skokih zelo specifična in že naučeni model na zbirki MPII [6] ni dovolj za dobro detekcijo delov telesa. Metodi CPM smo dodali podporo tudi za podatkovno zbirko COCO [7], ki se je obnesla nekoliko boljše, zato smo jo uporabili kot inicializacijo nevronske mreže za učenje na naši podatkovni zbirki. Metodi CPM smo morali dodati podporo za zbirko COCO, saj jo javno dostopna programska koda ne podpira, kajti zbirka ob izdaji članka še ni bila na voljo. Metoda CPM uporablja svojo različico programskega ogrodja Caffe [44], saj implementira poseben podatkovni sloj, ki sprejme poseben vhod. Vhod je sestavljen iz slike in dodatnega kanala slike, kjer so zapisani metapodatki, kot so npr. lokacije delov telesa.

Za pridobitev večjega števila učnih podatkov vhodne slike dodatno spremenimo (ang. augmentation) z množico enostavnih operacij za obdelavo slik. Metoda izvaja spremembe ob samem učenju, tako da so implementirane v sklopu programskega ogrodja Caffe. Implementiramo naslednje spremembe vhodnih podatkov:

- skaliranje vhodne slike (0.7-1.3),
- rotacija ( $\pm 40^\circ$ ),
- premik centra ( $\pm 40$  px),
- vertikalno zrcaljenje.

Slika 4.2a predstavlja primer vhodnega učnega podatka v metodo CPM. Arhitektura metode zahteva vhodno velikost 368 x 368, kjer za izhodišče vzamemo center detekcije smučarskega skakalca. Detekcija nam pomaga določiti tudi velikost skakalca. V delu nastavimo začetno velikost smučarskega skakalca na 70 % velikosti vhoda v arhitekturo. Slika 4.2b predstavlja izhod slike po opravljenih spremembah. Za posamezno spremembo podatkov določimo verjetnost, da se ta opravi. Na vhodni sliki se tako izvede naključno število sprememb z naključno vrednostjo posamezne spremembe v danem območju.

Kot smo že omenili v opisu uporabljenih metod, smo za detekcijo smučič smučarskega skakalca prav tako uporabili metodo CPM, tako da smo posamezne dele smučič obravnavali kot dodatne dele telesa. Zaradi spremenjene arhitekture je bilo treba učenje ostalih delov telesa na večjih zbirkah podatkov ponoviti. Tako smo dobili naučen model, ki zna detektirati dele telesa. Ta model smo nato uporabili za učenje na naši zbirki podatkov. Model za inicializacijo smo naučili na podatkovni zbirki COCO [7]. Model smo naučili s parametri, ki jih uporabljajo avtorji metode, in sicer tako dolgo, da se je vsaka vhodna slika spremenila 100-krat.



(a)

(b)

Slika 4.2: Sprememba vhodnih podatkov: (a) vhodna in (b) izhodna slika.

### 4.3 Metoda za ocenjevanje sloga smučarskega skoka

Vhod v metodo za ocenjevanje sloga smučarskega skoka predstavljajo lokacije delov telesa in smuči. Vhodne lokacije najprej normaliziramo. Prvi način normalizacije predstavlja uporaba masnega središča skakalca v posameznem slikovnem okvirju. Masno središče izračunamo kot uteženo povprečje lokacij delov telesa, pri čemer za uteži uporabimo zanesljivosti detekcij posameznih delov telesa. Za drugi način normalizacije uporabimo center detekcije smučarskega skakalca.

Detektirane trajektorije tudi spreminjamo tako, da pomikamo njihovo lokacijo. Premikamo celotno trajektorijo in tudi po posamezni sliki. Kot je razvidno s slike 3.9, predstavljajo veliko težavo odstopajoče meritve, ki so posledica slabših detekcij. Videoposnetki, ki jih uporabljamo, so prilagojeni televizijskemu prenosu in pridobljeni s spletne platforme YouTube. V tako pridobljenih posnetkih je predvsem zaradi počasne kamere in specifičnega kodiranja videoposnetka prisotno veliko število slikovnih okvirjev slabše ločljivosti. Da bi ta vpliv omilili, uporabljamo glajenje z mediano.

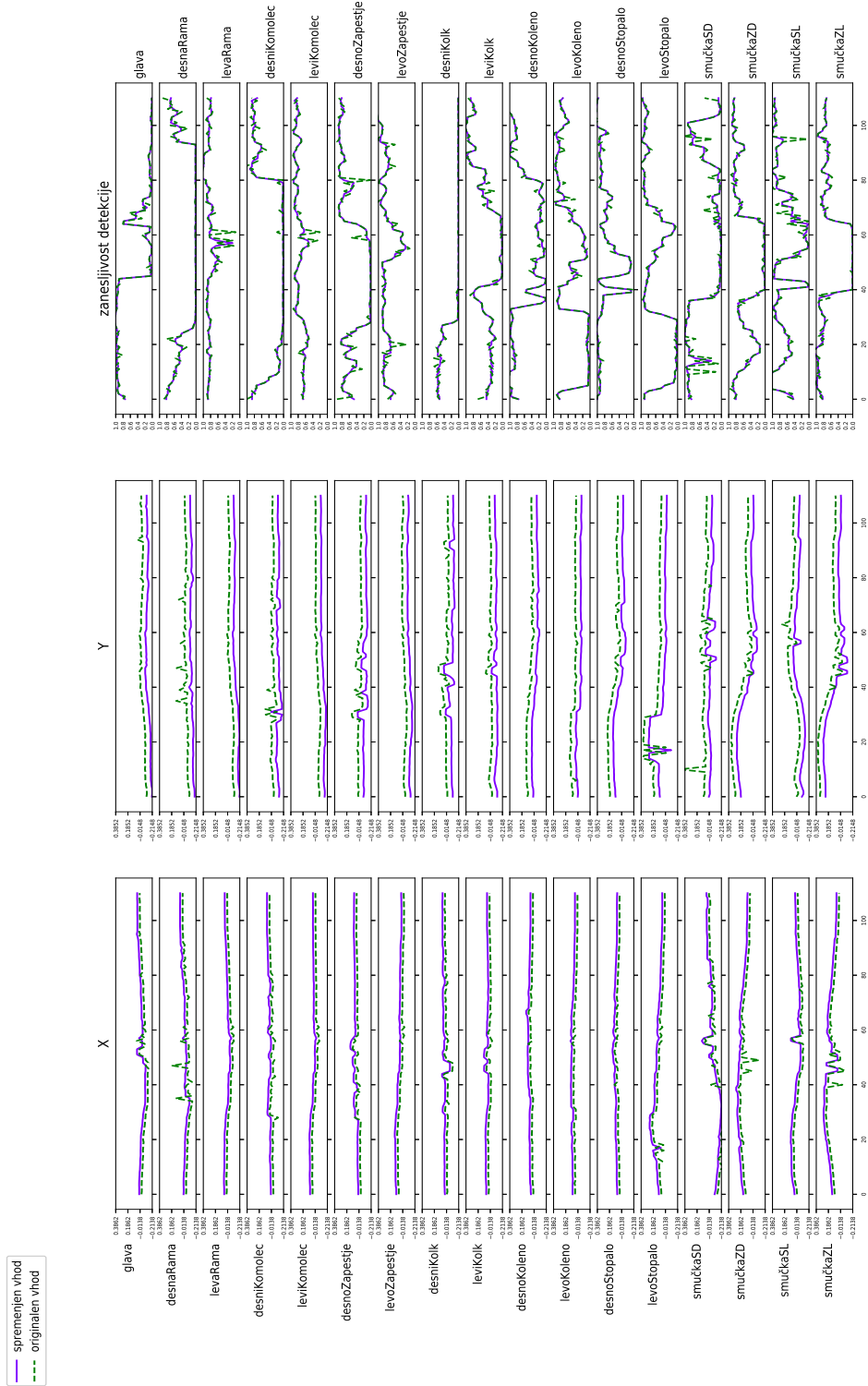
Uporabljamo različne velikosti filtra, s čimer želimo izničiti posamezne nepravilne detekcije. Pri glajenju je potrebno paziti, da ne izničimo posameznih podrobnosti, ki bi vplivale na sodniško oceno in niso napačne detekcije. Z ustrezno postavitvijo kamer, bi se lahko naučili transformacije v kanonični koordinatni sistem, s čimer bi lahko metodo posplošili na različna prizorišča tekem. Primer spremenjenega vhodnega podatka je prikazan na sliki 4.3. Skozi celoten skok naključno premaknemo detektirano masno središče, s čimer pozkušamo modelirati šum v trajektorijah, ki nastane kot posledica slabih detekcij. Vhodne trajektorije tudi filtriramo z mediano.

Za pridobitev sodniških ocen uporabljamo regresijsko arhitekturo, ki smo jo opisali v 3.4.2. Tako dobimo napovedi ocen za vsakega od petih sodnikov.

Za virtualno oceno smučarskega skoka, ki jo napove metoda, najprej odstranimo največjo in najmanjšo napovedano oceno. Preostale ocene nato povprečimo in zaokrožimo na najbližjo polovico, s čimer pridobimo virtualno sodniško oceno  $u_v$ .

Za ovrednotenje metode merimo konsistentost z ostalimi sodniškimi ocenami. Če postopek opišemo z enačbami, naj bodo  $\hat{y} = (y_1, y_2, y_3, y_4, y_5)$  anotirane sodniške ocene za skok nekega skakalca. Najprej odstranimo minimalno in maksimalno oceno ter nato povprečimo preostale ocene treh sodnikov, da dobimo  $\bar{y}$ , ki predstavlja referenčno oceno. Za posameznega sodnika nato izračunamo absolutno razdaljo  $d_i = |y_i - \bar{y}|$ , ki nam pove stopnjo konsistentnosti s preostalimi sodniki. Tako pridobimo konsistentnost anotiranih sodniških odločitev. Izračunamo še  $d_v = |u_v - \bar{y}|$ , ki nam opiše konsistentnost virtualne sodniške ocene  $u_v$  s preostalimi sodniki. Vrednosti povprečimo skozi vse skoke na tekmi, da dobimo povprečno odstopanje posameznega sodnika. Želimo si, da je napaka virtualnega sodnika podobna ali boljša od napak ostalih pravih sodnikov.

Kot najbolj enostavno mero uspešnosti uporabljamo tudi povprečno absolutno napako, ki jo izračunamo tako, da izračunamo absolutno razliko med dejanskimi ocenami sodnikov za posamezen skok in našimi napovedanimi ocenami. Napako nato povprečimo skozi vse skoke na tekmi.



**Slika 4.3:** Vizualizacija spremenjenega vhoda v metodo za ocenjevanje sloga smučarskega skoka. Slika prikazuje primer naključno premaknjene trajektorije skoka za vektor premika  $[-80, 96]$ . Premaknjena trajektorija je tudi glajena z mediano, in sicer z velikostjo filtra  $n = 3$ .



#### *4.3. METODA ZA OCENJEVANJE SLOGA SMUČARSKEGA SKOKA*49

Rezultate predstavimo v poglavju 6, še prej pa v naslednjem poglavju predstavimo podatkovno zbirko smučarskih skokov.



## Poglavje 5

# Podatkovna zbirka smučarskih skokov

V tem poglavju predstavimo našo podatkovno zbirko smučarskih skokov, ki smo jo uporabili v magistrskem delu. Najprej predstavimo obstoječe zbirke anotiranih podatkov za detekcijo delov telesa. Izkazalo se je, da z obstoječimi zbirkami anotiranih oseb ni mogoče doseči dobrih rezultatov na domeni smučarskih skokov, kar je posledica specifičnega modela interakcij med posameznimi deli telesa ter prisotnostjo smuči. Specifična podatkovna zbirka je bila potrebna tudi zaradi detekcije smuči z uporabo metode za detekcijo delov telesa. Poleg anotacij delov telesa in smuči vsebuje podatkovna zbirka tudi podatke o sami tekmi, ki jih uporabimo za napoved sodniških ocen.

### 5.1 Obstoječe zbirke podatkov

**Leeds Sports Pose Dataset (LSP)** [20] predstavlja eno izmed prvih zbirk podatkov, ki se danes še vedno uporablja za ovrednotenje novih metod. Zbirka je sestavljena iz 2000 anotiranih slik oseb predvsem s športne domene. Anotiran je celotni del telesa. Zbirka je bila sestavljena s pomočjo slik, pridobljenih s spletnega servisa Flickr, ki za posamezne slike vsebuje

anotacije. Zbirka je razdeljena na 1000 učnih in 1000 testnih slik. Kasneje so avtorji predstavili še razširjeno zbirko podatkov, ki vsebuje 10.000 anotiranih oseb [21].

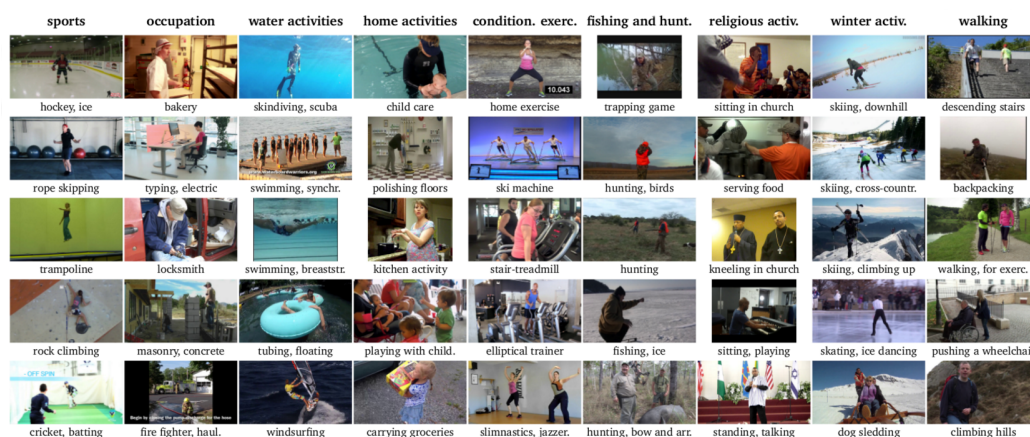
**Frames Labeled In Cinema (FLIC)** [22] vsebuje 5003 anotiranih slik, zbranih iz različnih filmov. Slike iz posameznih filmov so bile pridobljene s pomočjo detektorja oseb, s katerim so detektirali osebe v vsakem desetem slikovnem okvirju celotnega filma. Za gradnjo zbirke so uporabili zbirko 30 filmov. Tako detektiranih oseb z dovolj visoko zanesljivostjo je bilo 20.000. Za anotacijo delov telesa so uporabili platformo Amazon Mechanical Turk, saj omogoča vključitev širše množice ljudi, ki za majhno plačilo anotirajo posamezne slike. Anotiran je zgolj zgornji del telesa. Zbirka je razdeljena na 3987 učnih in 1016 testnih slik. Primeri anotacij so prikazani na sliki 5.1.



**Slika 5.1:** Primer podatkov iz podatkovne zbirke FLIC [22].

**MPII Human Pose Dataset (MPII)** [6] zbirka podatkov predstavlja drugo največjo zbirko anotiranih podatkov za detekcijo delov telesa. Zbirka je zgrajena okoli dvonivojske hierarhije človekovih aktivnosti. Hierarhija predstavlja 823 aktivnosti, razvrščenih v 21 kategorij. Primeri kategorij so prikazani na sliki 5.2. Tako organizirana zbirka omogoča boljšo ovrednotenje metod, saj lahko ovrednotenje opravimo po posameznih domenah. Tako sestavljena zbirka tudi bolje pokrije različne razporeditve delov telesa, ki se pojavljajo v realnih primerih. Slike so pridobljene s platforme YouTube s pomočjo zahtevkov, ki so vsebovali opise posameznih aktivnosti. Zbirka vsebuje približno 28.000 učnih podatkov in 12.000 testnih.

**COCO Keypoints Challenge (COCO)** [7] v sklopu svojega tekmovanja predstavlja največjo zbirko anotiranih podatkov. Zbirka vsebuje več kot



**Slika 5.2:** Primer podatkov razvrščenih v različne kategorije aktivnosti podatkovne zbirke MPII [6].

140.000 anotiranih oseb, kar predstavlja skupno več kot milijon anotiranih delov telesa. Zbirka je razdeljena na učno (100.000 oseb), validacijsko (20.000 oseb) in testno (20.000 oseb) množico. Testna množica ni dostopna. Tekmovanje COCO zahteva detekcijo delov telesa več oseb, pri čemer lokacije posameznih oseb niso znane in jih mora metoda ustrezno detektirati. Primeri anotacij so prikazani na sliki 5.3.



**Slika 5.3:** Primer podatkov iz podatkovne zbirke COCO [7].

## 5.2 Podatkovna zbirka smučarskih skokov

Podatkovna zbirka smučarskih skokov predstavlja prvo tovrstno zbirko in je namenjena metodam za detekcijo delov telesa in smuči. Vsebuje tudi anotacije posameznih sodniških ocen in dolžin skokov. Zbirka vsebuje skoraj 1800 anotiranih slik. Primeri slik so predstavljeni na sliki 5.4.



**Slika 5.4:** Primeri slik s podatkovne zbirke smučarskih skokov.

Pri gradnji zbirke smo uporabili tekme z olimpijskih iger v Vancouvru 2010. Uporabili smo tekme na veliki in mali skakalnici. Za te tekme smo se odločili, ker so bili posnetki tekem v polni HD resoluciji na voljo na spletu. Pomembna je tudi razporeditev kamer, saj mora biti ta čimbolj konsistentna med posameznimi tekmami, obenem pa mora biti skakalec vedno dobro viden. V realnem sistemu bi za tak sistem uporabili posebne kamere, razporejene po izbranih delih skakalnice.

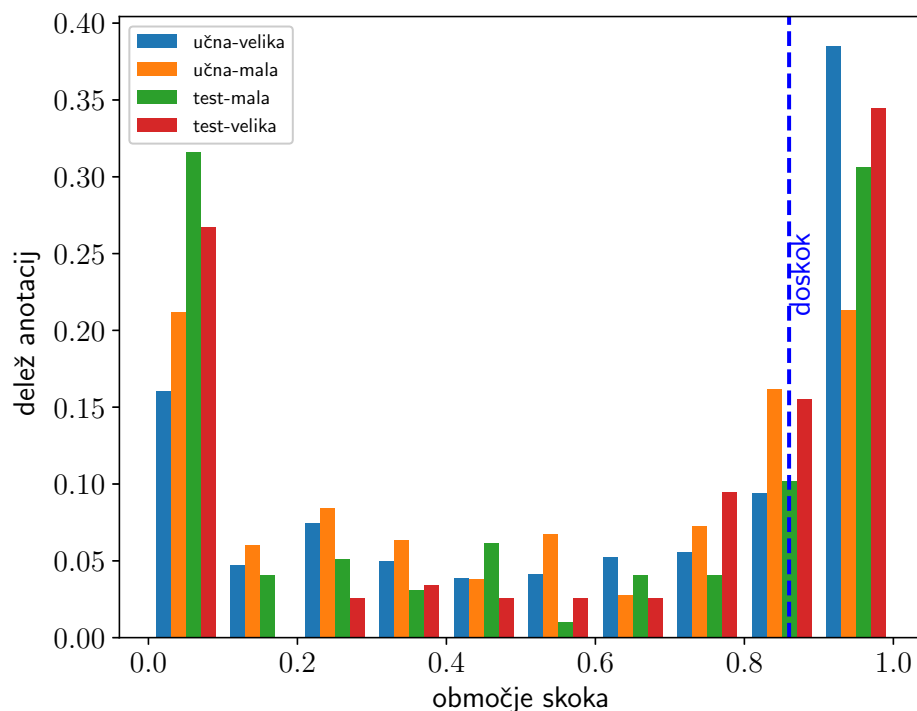
### 5.2.1 Anotacije delov telesa in smuči

Za anotacijo delov telesa smo uporabili tekmi na veliki in mali skakalnici. Zbirka je sestavljena iz 1761 anotiranih slik smučarskih skakalcev. Razporeditev anotacij po posamezni skakalnici je predstavljena v tabeli 5.1. Učni del zbirke je bil zgrajen na skokih prve serije tekme, testni del pa na skokih finalne serije.

**Tabela 5.1:** Razporeditev anotacij po posamezni skakalnici in delu zbirke.

skakalnica	učna množica	testna množica	$\Sigma$
velika	679	116	795
mala	868	98	966
$\Sigma$	1547	214	1761

Histogram na sliki 5.5 predstavlja razporeditev anotacij po delu skoka. Start skoka je definiran kot 5. slikovni okvir po odskoku, zaključek skoka pa 15. slikovni okvir po doskoku. Za tako definicijo skoka smo se odločili zaradi omejitve posnetkov tekem, saj so slikovni okvirji ob odskoku preveč popačeni, da bi jih lahko uporabili. Pri doskoku pa je prisotna težava zaradi menjave kamere, saj smo si želeli, da je za potrebe kasnejše uporabe podatkov detekcij za napoved sodniških ocen celoten skok zajet v eni kameri. Na histogramu je zato zajet del zbirke, ki je v območju tako definiranega skoka in ga uporabimo za napoved sodniških ocen. Takih slik je 1156, ostale anotirane slike so bile zajete izven samega skoka, npr. v počasnem posnetku, vožnji po vzletišču ali drugih delih skoka, s čimer smo želeli izboljšati samo delovanje detektorja delov telesa in smuči. Opazimo, da je anotacij največ v delu po odskoku in pri doskoku, saj je v tem delu skoka prisotno največ gibanja. V fazi leta je pozicija telesa dokaj fiksna in ne potrebujemo tako veliko anotacij. Anotacije za posamezni slikovni okvir smo zbrali v json formatu, ki je predstavljen na sliki 5.6. Anotirali smo 13 delov telesa in smuči, kot je to prikazano na sliki 5.7.



**Slika 5.5:** Razporeditev anotacij po delih skoka. Skok je na histogramu razdeljen na deset delov.

### 5.2.2 Anotacije smučarskih tekem

Anotacije smučarskih tekem so potrebne za napoved sodniških ocen. V magistrskem delu anotiramo posamezni skok s skakalcem (koda FIS), njegovo dolžino in sodniškimi ocenami. V magistrskem delu smo se tudi za ocenjevanje sloga smučarskih skokov osredotočili na tekme olimpijskih iger v Vancouvru, in sicer predvsem zaradi delovanja detektorja delov telesa, saj je naučen zgolj iz podatkov s teh tekem. Za splošno delovanje bi potrebovali anotacije z različnih prizorišč. Težavo pri tako splošnem pristopu predstavlja pridobitev posnetkov ter njihova anotacija, ki je časovno zelo zamudna.



**Slika 5.6:** Primer anotacije slikovnega okvirja v json formatu.

```
{
  "149": {
    "head": [
      1100.25,
      247.34,
      48.61,
      44.05
    ],
    "rect": [
      874.25,
      197.75,
      598.50,
      586.50
    ],
    "validation": true,
    # vidnost posameznih delov telesa
    "visible": [1, 0, 1, ...],
    "x": [
      1123.25,
      0,
      1099.25,
      ...
    ],
    "y": [
      272.75,
      0,
      277.25,
      ...
    ]
  }
}
```



**Slika 5.7:** Primer anotiranega slikovnega okvirja.

Z olimpijskih iger v Vancouvru smo anotirali naslednje tekme:

- tekma smučarskih skakalcev na veliki skakalnici,
- tekma smučarskih skakalcev na mali skakalnici,
- kvalifikacijska tekma smučarskih skakalcev na veliki skakalnici,
- tekma kombinatorcev na mali skakalnici.

V poglavju 6 predstavimo rezultate uporabljenih metod.

## Poglavje 6

### Rezultati

V tem poglavju predstavimo glavne rezultate magistrskega dela. Najprej predstavimo rezultate metode za detekcijo smučarskega skakalca, ki predstavlja vhod v metodo za detekcijo delov telesa in smuči, katere rezultate ravno tako predstavimo. Obe metodi ovrednotimo na naši testni zbirki podatkov. Za ovrednotenje detekcij smučarskega skakalca uporabljamo mero uspešnosti, ki jo uporabljajo na zbirki podatkov PASCAL VOC [37]. Rezultate detekcij delov telesa in smuči predstavimo v obliki, ki jo uporabljajo na evaluacijski zbirki MPII [6]. V nadaljevanju nato predstavimo rezultate napovedovanja sodniških ocen na tekmah olimpijskih iger v Vancouvru 2010. Podrobno predstavimo vhode v metodo za ocenjevanje sloga smučarskega skoka ter vpliv različnih metod procesiranja vhodnih podatkov na izboljšanje rezultatov. Z izbranimi merami uspešnosti, ki jih uporabljamo za ovrednotenje kakovosti napovedi sodniških ocen, ovrednotimo naše napovedi. Doseženi rezultati pokažejo, da je mogoče z našo metodo napovedovati sodniške ocene, ki so konsistentne z ocenami ostalih sodnikov.

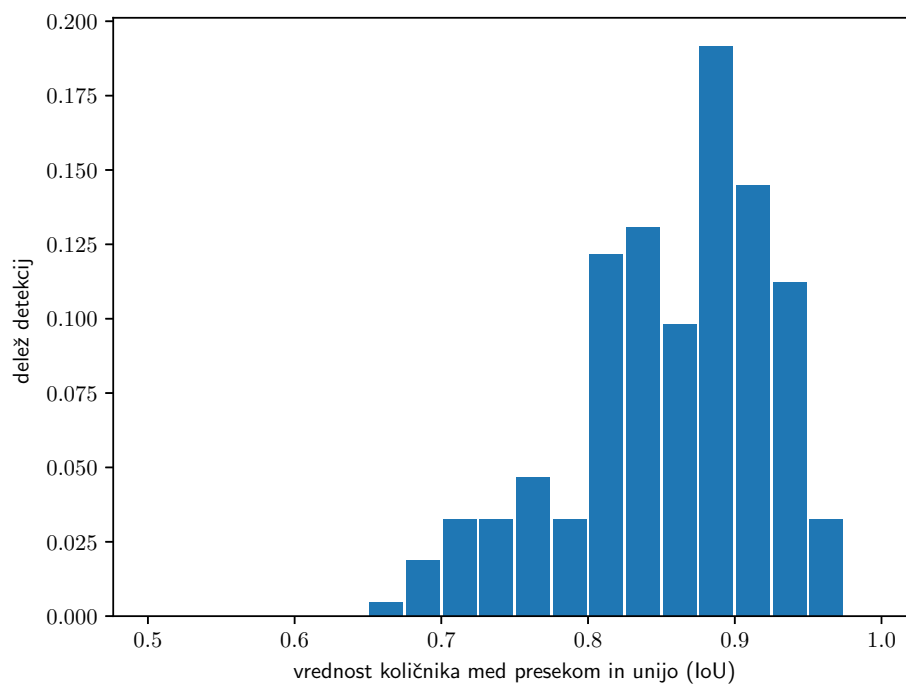
## 6.1 Detekcija smučarskega skakalca

Metodo za detekcijo smučarskega skakalca ovrednotimo na naši testni zbirki podatkov, ki je sestavljena iz tekme na veliki in mali skakalnici olimpijskih iger v Vancouvru. Kot mero uspešnosti uporabimo količnik med presekom in unijo (IoU) anotiranega in napovedanega okvirja detekcije. Metrika je definirana z enačbo (6.1), kjer  $|A|$  predstavlja površino okvirja A. Najbolj poznana je uporaba metrike na priznanem tekmovanju za detekcijo objektov PASCAL VOC [37].

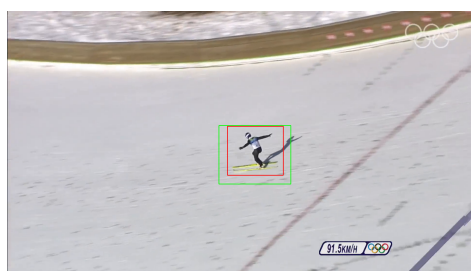
$$IoU = \frac{|refBB_{\text{Box}} \cap predBB_{\text{Box}}|}{|refBB_{\text{Box}} \cup predBB_{\text{Box}}|} \quad (6.1)$$

Rezultati detekcij na testni množici so predstavljeni na histogramu 6.1. Histogram prikazuje razporeditev vrednosti količnika med presekom in unijo (IoU). Opazimo, da je več kot 80 % detekcij detektiranih z vrednostjo  $IoU \geq 0.8$ . Vrednost  $IoU = 1$  pomeni idealno detekcijo, kar pomeni, da naš detektor zelo dobro detektira smučarskega skakalca. Primera najslabše in najboljše detekcije na naši testni zbirki sta predstavljena na sliki 6.2.

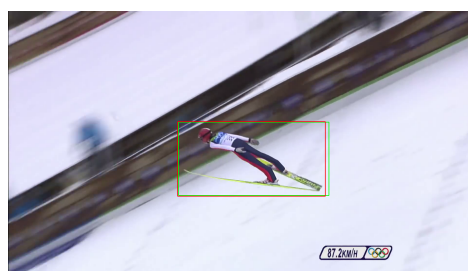
Kot vidimo, deluje detektor smučarskega skakalca odlično in v popolnosti zadosti zahtevam metode za detekcijo delov telesa in smuči, ki jo predstavimo v naslednjem podpoglavju.



**Slika 6.1:** Rezultati detekcije smučarskega skakalca.



(a)  $IoU = 0.65$



(b)  $IoU = 0.96$

**Slika 6.2:** Primer najslabše in najboljše detekcije smučarskega skakalca. Rdeč okvir prikazuje referenčno detekcijo, zelen okvir pa napovedano detekcijo.

## 6.2 Detekcija delov telesa in smuči

Detektor delov telesa in smuči ovrednotimo na testni zbirki podatkov smučarskih skokov. Testno zbirko smo anotirali na veliki in mali skakalnici. Skupno smo anotirali 214 slikovnih okvirjev, kot je to predstavljeno v tabeli 5.1. Za ovrednotenje uporabimo metriko PCK, ki je bila prvič predstavljena v delu [46]. Detekcija posameznega dela telesa je pravilna, če detekcija leži v oddaljenosti od pravilne lokacije, ki je manjša od  $\alpha * \max(h, w)$ , kjer  $h, w$  predstavlja višino in širino okvirja okoli osebe, katere dele telesa želimo detektirati.  $\alpha$  predstavlja mejno vrednost maksimalne razdalje. MPII [6] zbirka podatkov uporablja spremenjeno definicijo PCK metrike, ki jo uporabimo tudi v našem delu. PCKh, kot jo imenujejo, uporablja anotacijo glave namesto celotne osebe ter tako zmanjša vpliv različne razporeditve delov telesa. Maksimalno razdaljo, ki še smatra detekcijo za pravilno, opišemo z enačbo (6.2), kjer  $X_1$  in  $X_2$  predstavljata lokaciji diagonalno nasprotnih oglišč pravokotnika, ki obdaja glavo osebe. Posebnost enačbe je vrednost pristranskosti 0.6, ki jo uporabljajo v evalvaciji MPII, na katero se ob ponovnih implementacijah metrike velikokrat pozabi. MPII [6] za ovrednotenje uporablja vrednost  $\alpha = 0.5$ . Za različne parametre  $\alpha$  je metrika vizualno predstavljena na sliki 6.3.

$$\max\_dist = \alpha * 0.6 * \|X_1 - X_2\|_{L_2} \quad (6.2)$$

Graf 6.4 prikazuje vrednost PCKh za različne parametre  $\alpha$  za vse dele telesa in smuči. Opazimo, da je večina delov telesa in smuči pravilno detektirana v več kot 50 % primerih pri vrednosti parametra  $\alpha = 0.15$ , kar ustreza velikostnemu redu napake, ki je enak približno 10 % velikosti glave smučarskega skakalca, kar predstavlja zelo majhno napako. Opazimo tudi, da je desna polovica telesa detektirana nekoliko slabše. Najslabše je detektiran desni kolk, ki bistveno odstopa od povprečja, saj je dobro viden zelo redko, posledično pa je tudi zelo slabo zastopan v učni zbirki naše podatkovne zbirke. Vzrok je predvsem v poziciji kamere, saj je na izbranih tekmah ka-

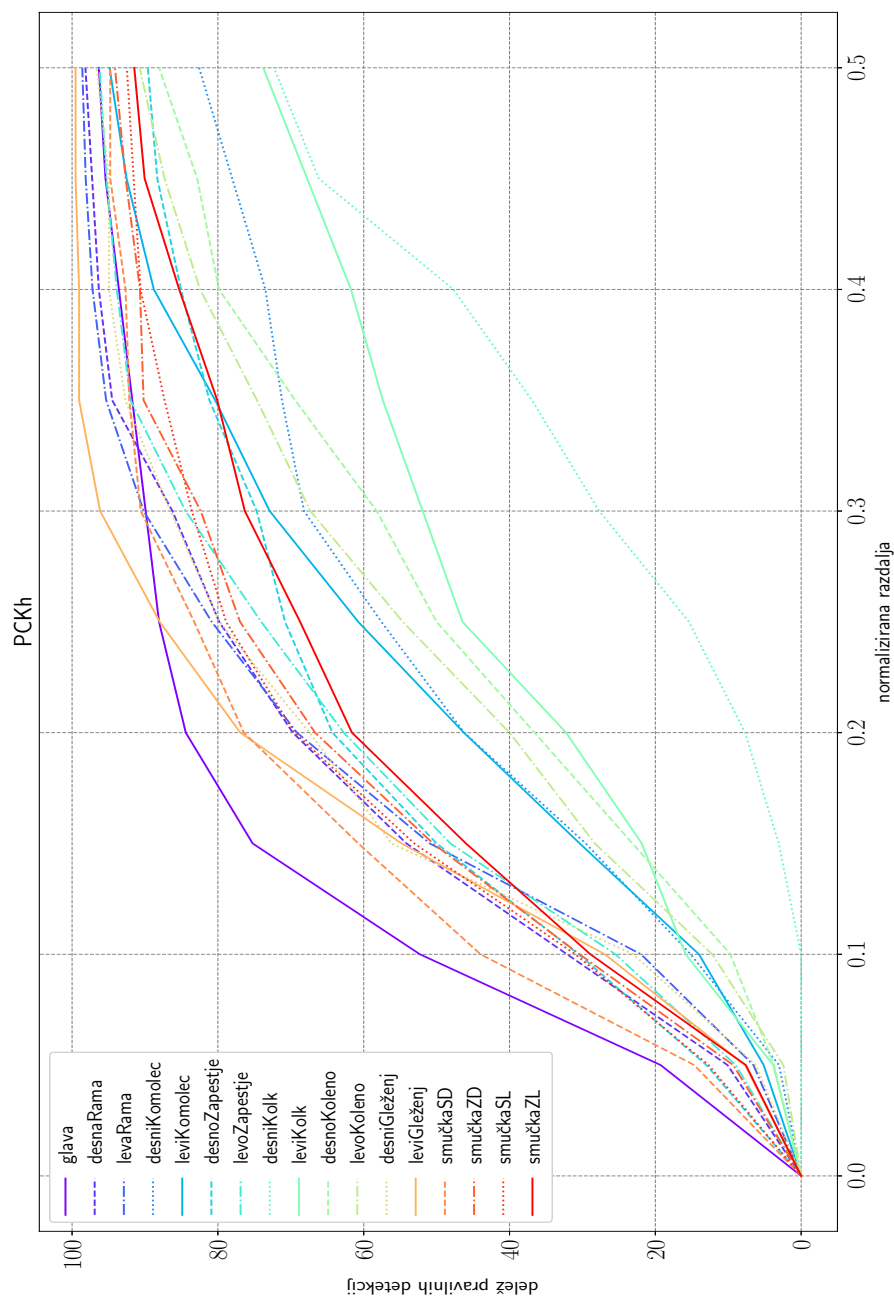


**Slika 6.3:** Vizualizacija metrike PCKh za vrednosti parametra  $\alpha$  od 0.1 do 0.5 s korakom 0.1.

mera postavljena na levi strani. Nekoliko manj natančno so detektirani tudi komolci in kolena, predvsem zaradi pomanjkanja teksture na dresu. Opazimo tudi, da je kakovost detekcij smuči podobna ali celo nekoliko boljša od kakovosti detekcij ostalih delov telesa, iz česar lahko sklepamo, da se metoda uspešno nauči detektirati dele smuči podobno kot preostale dele telesa.

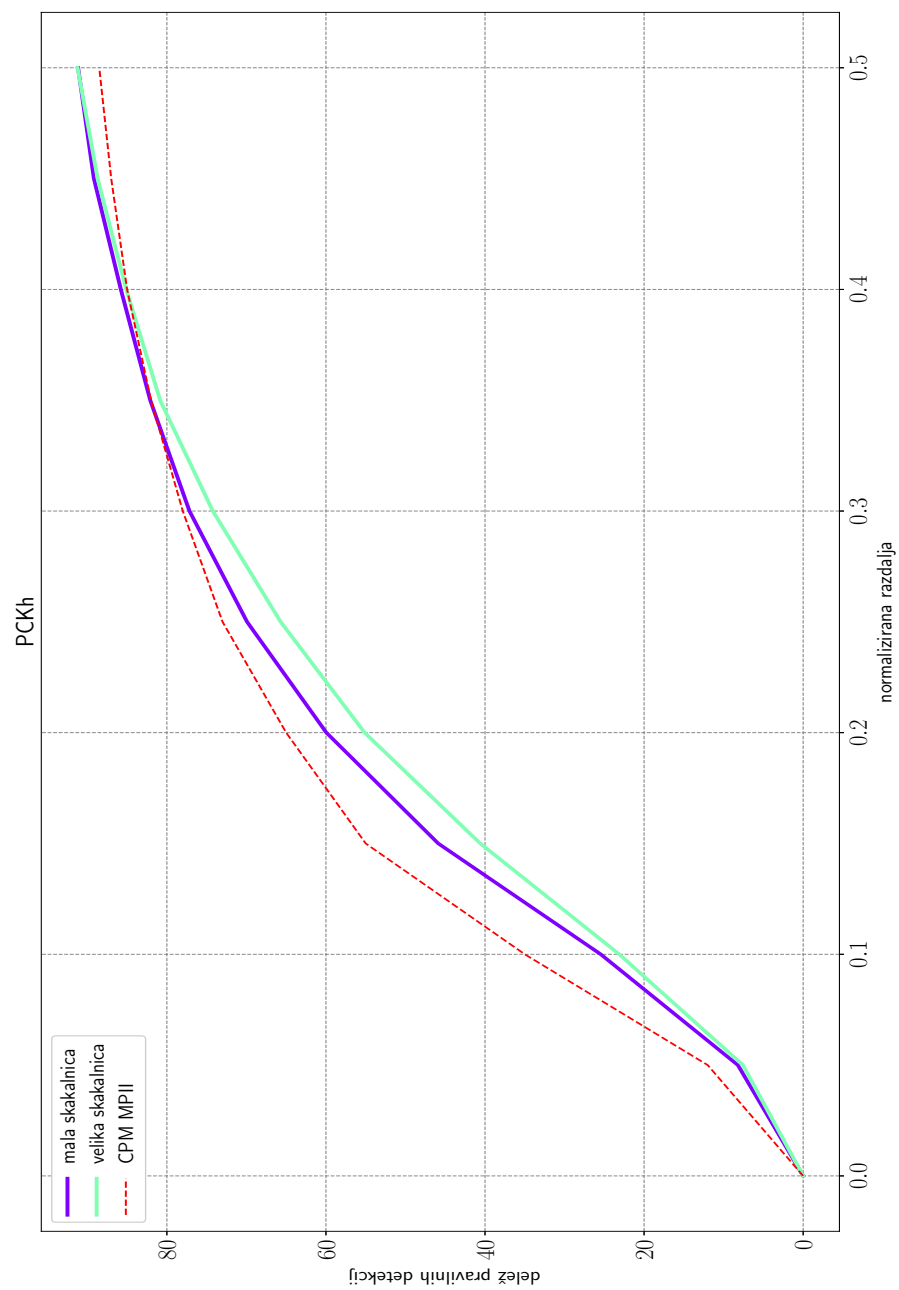
Graf 6.5 prikazuje povprečno vrednost PCKh, izračunano kot povprečje PCKh vrednosti po posameznih delih telesa (brez smuči). Opazimo, da so rezultati na mali skakalnici nekoliko boljši. Vzrok verjetno leži v nekoliko slabši kakovosti videoposnetkov na veliki skakalnici, saj je oddaljenost kamere nekoliko večja. Čeprav rezultati niso neposredno primerljivi, podamo tudi rezultate metode CPM [13] na podatkovni zbirki MPII [6]. Opazimo, da rezultati niso dosti slabši, saj dele telesa na domeni smučarskih skokov detektiramo s podobno natančnostjo.

Kvalitativni rezultati detekcij delov telesa in smuči so predstavljeni na sliki 6.6. Prikazani so vsi deli telesa in smuči, ki jih detektiramo. Tudi tukaj se opazi, da so desni deli telesa detektirani nekoliko slabše. Slabše so detektirani tudi deli telesa, ki so slabo vidni in tudi niso bili anotirani v učni zbirki. Takšne detekcije filtriramo s pomočjo zanesljivosti detekcije, katere vrednost pridobimo iz verjetnostnih map.



**Slika 6.4:** Vrednost PCKh za različne vrednosti  $\alpha$  za vse dele telesa in smuči.





**Slika 6.5:** Povprečna vrednost PCKh za različne vrednosti  $\alpha$  po posamezni skakalnici.



**Slika 6.6:** Primeri detekcij delov telesa in smuči. Deli telesa in smuči so prikazani po vrsticah, kjer je v primeru simetrije najprej prikazan desni del telesa ali smuči.

## 6.3 Ocenjevanje sloga smučarskega skoka

V tem delu predstavimo rezultate metode za ocenjevanje sloga smučarskega skoka. Metodo ovrednotimo na dveh tekmah olimpijskih iger v Vancouvru 2010. Za ovrednotenje metode uporabimo metrike, ki smo jih definirali v opisu implementacije metode 4.3.

Tabela 6.1 predstavlja referenčne in napovedane sodniške ocene finalne serije tekme na mali skakalnici po posameznih sodnikih. Vhodni podatki so bili normalizirani glede na masno središče skakalca v posameznem slikovnem okvirju. Model smo naučili zgolj na podatkih iz prve serije. Napovedane ocene so zaokrožene na najbližjo polovico. V tabeli so predstavljene tudi napovedi virtualnega sodnika. Zadnja vrstica tabele predstavlja povprečno sodniško oceno izračunano po posameznih sodnikih.

V tabeli 6.2 so predstavljene izračunane povprečne absolutne napake po posameznih sodnikih, za katere napovemo sodniško oceno.  $x$  predstavlja napako brez zaokroževanja ocen na najbližjo polovico,  $\hat{x}$  napako z zaokrožitvijo na najbližjo polovico. Za primerjavo izračunamo tudi napako  $\bar{x}$ , ki jo naredimo, če za napoved uporabimo povprečje ocen iz prve serije tekme. Ugotovimo, da naša metoda napoveduje sodniške ocene približno dvakrat bolj natančno od enostavnega povprečnega sodnika.

Slika 6.7 prikazuje matriko zamenjav, pridobljeno z napovedmi ocen za obstoječe sodnike, ki so prikazane v tabeli 6.1. Opazimo, da je v večini primerov napaka napovedi največ polovica ocene. Opazimo tudi, da je pri manjših referenčnih ocenah naša napoved največkrat nekoliko višja; podobno pri višjih ocenah, kjer je naša napoved največkrat nekoliko nižja. Vzrok po našem mnenju najdemo v pomanjkanju robnih primerov v naši učni množici.

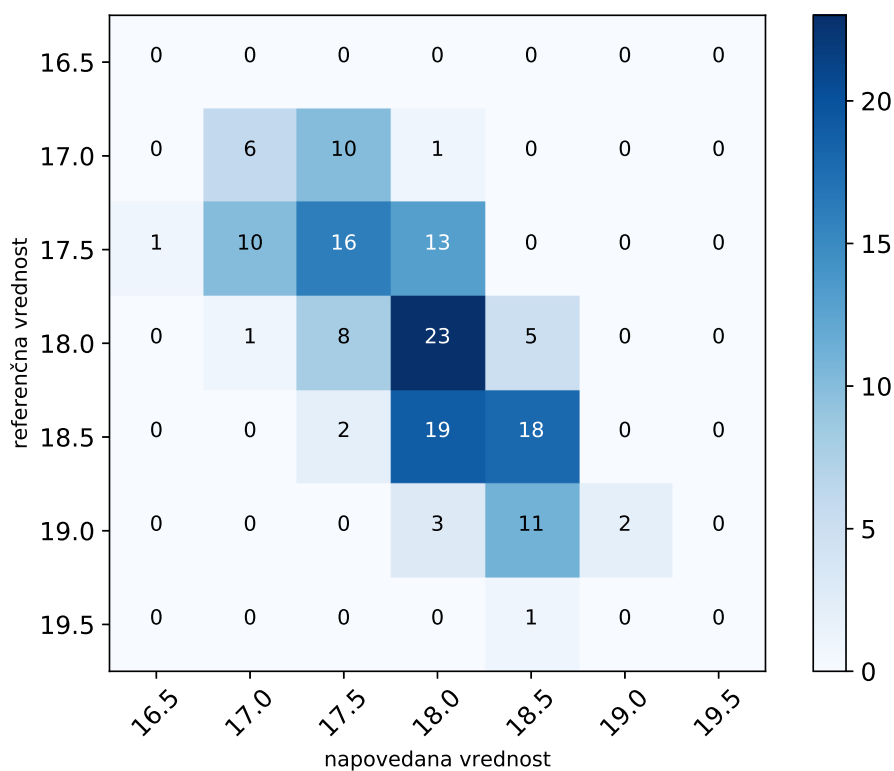
Tabela 6.3 prikazuje odstopanje ocen posameznih sodnikov od robustnega povprečja posameznega skoka, kjer smo predhodno odstranili najvišjo in najmanjšo oceno, kot smo to opisali v opisu implementacije metode 4.3. Opazimo, da ocena virtualnega sodnika odstopa nekoliko več od ostalih sodnikov. Želimo si, da bi bilo to odstopanje podobno ali manjše od ostalih sodnikov.

**Tabela 6.1:** Referenčne in napovedane sodniške ocene (dejanska/napovedana) finalne serije na mali skakalnici.

skakalec	sodnik 1	sodnik 2	sodnik 3	sodnik 4	sodnik 5	virtualni sodnik
1	17.5/17.5	17.5/17.0	17.0/17.5	17.5/17.0	17.0/17.5	17.5/17.5
2	17.5/18.0	17.5/17.0	17.5/17.5	17.5/17.5	17.5/17.0	17.5/17.5
3	17.5/18.0	17.5/17.5	17.0/17.5	17.0/17.0	17.5/17.5	17.5/17.5
4	18.0/18.0	17.5/17.5	17.5/18.0	17.0/17.5	18.0/18.0	17.5/18.0
5	18.5/18.0	18.0/17.5	18.0/18.0	17.5/18.0	18.0/18.0	18.0/18.0
6	17.5/18.0	17.5/17.5	17.0/17.5	17.0/17.0	17.5/17.5	17.5/17.5
7	17.0/17.5	17.0/17.5	17.0/17.5	17.0/17.5	17.0/17.5	17.0/17.5
8	18.5/18.0	18.0/17.5	18.0/17.5	17.5/17.5	18.0/17.5	18.0/17.5
9	17.5/17.0	17.0/17.0	17.5/16.5	17.5/17.0	17.5/17.0	17.5/17.0
10	17.5/17.5	17.0/17.0	17.0/17.0	17.0/17.0	17.5/17.5	17.0/17.0
11	18.0/18.0	17.5/17.5	18.0/18.0	17.5/18.0	17.5/18.0	17.5/18.0
12	18.0/17.5	17.5/17.0	17.5/17.0	17.5/17.0	17.5/17.5	17.5/17.0
13	18.0/18.0	17.5/17.5	17.0/18.0	17.5/18.0	17.5/18.0	17.5/18.0
14	18.0/18.0	17.5/18.0	18.0/18.5	17.5/18.0	18.0/18.5	18.0/18.0
15	19.0/18.5	18.5/18.5	18.5/18.5	18.0/18.0	19.0/18.5	18.5/18.5
16	19.0/18.0	18.5/18.0	18.5/18.0	18.5/18.0	18.5/18.5	18.5/18.0
17	19.0/19.0	18.5/18.5	18.5/18.5	18.0/18.5	19.0/18.5	18.5/18.5
18	18.5/18.0	18.0/18.0	18.0/18.0	18.5/18.0	18.5/18.0	18.5/18.0
19	18.5/18.5	18.5/18.5	18.5/18.5	18.0/18.0	19.0/18.5	18.5/18.5
20	17.5/18.0	17.5/17.5	18.0/18.0	17.5/17.5	18.0/17.5	17.5/17.5
21	18.5/18.0	18.0/17.5	18.5/18.0	18.0/17.0	18.0/17.5	18.0/17.5
22	18.0/18.0	18.0/18.0	18.0/18.0	18.0/18.0	18.0/18.0	18.0/18.0
23	18.5/18.0	18.0/18.0	18.0/18.0	18.0/18.0	18.0/18.0	18.0/18.0
24	19.0/18.5	19.0/18.5	19.0/18.5	19.0/18.5	19.5/18.5	19.0/18.5
25	18.5/18.0	18.5/17.5	18.5/18.0	18.5/17.5	18.5/18.0	18.5/18.0
26	18.5/18.5	18.5/18.5	18.5/18.5	18.0/18.5	18.5/18.5	18.5/18.5
27	18.5/18.0	18.5/18.0	19.0/18.0	18.5/18.5	18.5/18.5	18.5/18.0
28	18.5/18.5	19.0/18.5	19.0/18.5	19.0/18.0	19.0/19.0	19.0/18.5
29	18.5/18.5	18.5/18.0	18.5/18.5	18.0/18.0	19.0/18.5	18.5/18.5
30	18.5/18.5	18.5/18.0	17.5/18.0	18.5/18.0	18.0/18.5	18.5/18.5
$\bar{x} \pm \sigma$	18.18 $\pm$ 0.54/ 18.05 $\pm$ 0.39	17.95 $\pm$ 0.57/ 17.75 $\pm$ 0.50	17.95 $\pm$ 0.64/ 17.92 $\pm$ 0.50	17.82 $\pm$ 0.57/ 17.73 $\pm$ 0.50	18.10 $\pm$ 0.64/ 17.98 $\pm$ 0.51	18.00 $\pm$ 0.55/ 17.90 $\pm$ 0.47

**Tabela 6.2:** Povprečna absolutna napaka in standardni odklon po posameznih sodnikih na mali skakalnici brez dodatnega procesiranja vhodnih podatkov.

Vrsta napovedi	sodnik 1	sodnik 2	sodnik 3	sodnik 4	sodnik 5
$x \pm \sigma$	$0.30 \pm 0.21$	$0.31 \pm 0.23$	$0.34 \pm 0.30$	$0.39 \pm 0.27$	$0.31 \pm 0.19$
$\hat{x} \pm \sigma$	$0.30 \pm 0.28$	$0.27 \pm 0.28$	$0.33 \pm 0.32$	$0.35 \pm 0.32$	$0.32 \pm 0.27$
$\bar{x} \pm \sigma$	$0.60 \pm 0.35$	$0.66 \pm 0.48$	$0.69 \pm 0.44$	$0.54 \pm 0.44$	$0.64 \pm 0.52$



**Slika 6.7:** Matrika zamenjav, pridobljena z napovedjo ocen obstoječih sodnikov v tabeli 6.1.

**Tabela 6.3:** Povprečna absolutna napaka konsistentnosti z ostalimi sodniki.

sodnik 1	sodnik 2	sodnik 3	sodnik 4	sodnik 5	virtualni sodnik
$0.22 \pm 0.19$	$0.10 \pm 0.13$	$0.17 \pm 0.21$	$0.22 \pm 0.23$	$0.16 \pm 0.17$	<b><math>0.27 \pm 0.21</math></b>

Za izboljšanje delovanja uporabimo različne metode procesiranja vhodnih podatkov, ki smo jih predstavili v opisu implementacije metode 4.3. Iz zanesljivosti detekcij opazimo, da se v večini primerov ob slabi zanesljivosti poslabšajo tudi lokacije delov telesa, saj vsebujejo več šuma. Za odstranitev šuma uporabimo glajenje z mediano z velikostjo filtra  $n = 3$ . Glajenje skozi več slikovnih okvirjev poslabša kakovost napovedovanja ocen, saj odstranimo preveč podatkov, ki vplivajo na oceno skoka. Rezultati so predstavljeni v tabeli 6.4. Opazimo, da samo glajenje ne izboljša bistveno delovanja, saj z odstranitvijo šuma odstranimo tudi majhne podrobnosti, ki bi drugače pravilno vplivale na oceno skoka.

**Tabela 6.4:** Vpliv glajenja na napako konsistentnosti.

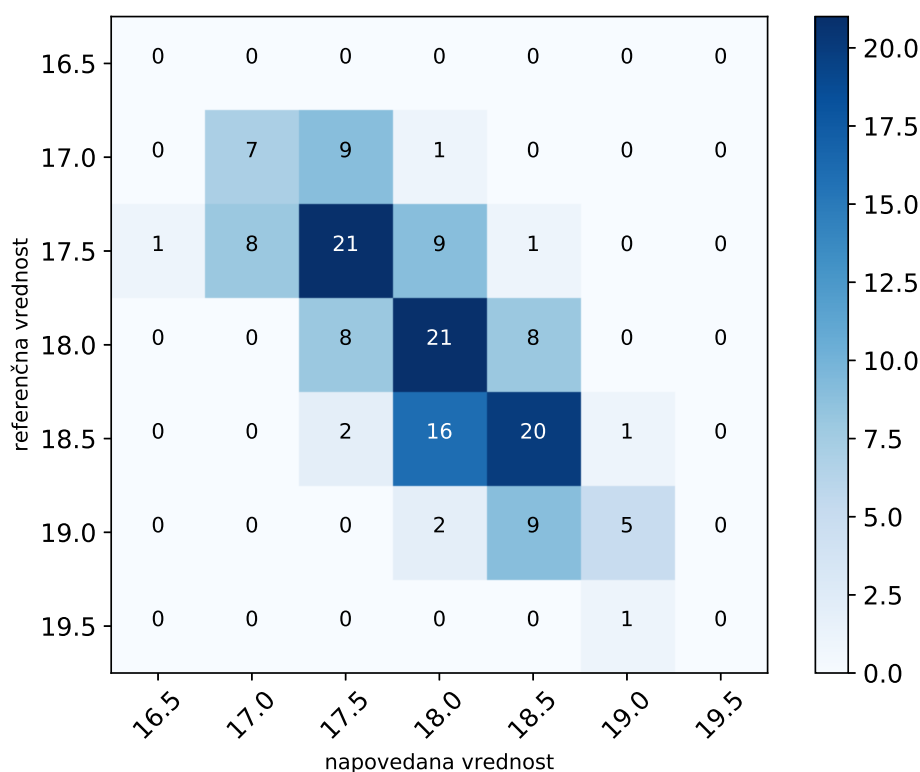
n	virtualni sodnik
0	$0.27 \pm 0.21$
3	$0.25 \pm 0.24$
5	$0.29 \pm 0.25$
25	$0.37 \pm 0.29$

Ugotovimo tudi, da je podatek o zanesljivosti detekcij delov telesa zelo pomemben, saj se model sam v veliki meri nauči filtriranja slabih podatkov in predstavlja najboljši način filtriranja vhodnih podatkov. Če izpustimo podatek o zanesljivosti detekcij, dosežemo vrednost napake konsistentnosti  **$0.39 \pm 0.33$** .

Ovrednotili smo tudi vpliv spremembe (ang. augmentation) trajektorij. Uporabimo 10 sprememb vhodnih podatkov s premikom centra v območju  $[-100, 100]$ . Podobno kot pri glajenju z mediano tudi tukaj ugotovimo, da izbrane spremembe vhodnih podatkov ne pripomorejo k statistično pomembno boljšim rezultatom.

Najboljši rezultat dosežemo, če v učno množico vključimo več podatkov. Če dodamo v učno množico tekmo kombinatorcev, ki se odvija na isti skakalnici in tako pridobimo dodatnih 50 učnih primerov, dosežemo vrednost napake konsistentnosti  **$0.21 \pm 0.20$** . Matrika zamenjav za ta model je pred-

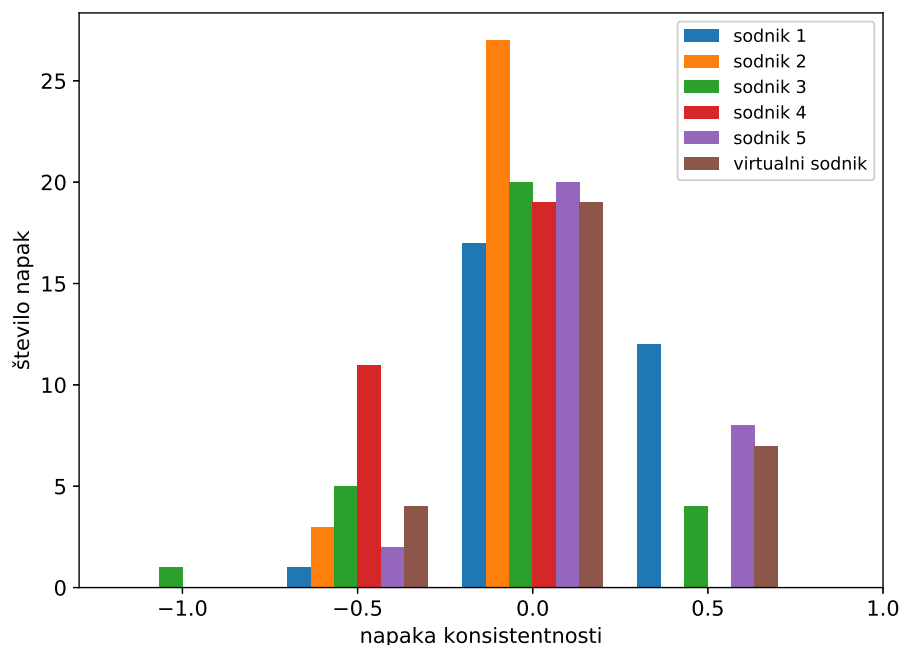
stavljena na sliki 6.8. Izboljšanje delovanja opazimo predvsem pri robnih primerih, saj je povprečna ocena kombinatorcev za skoraj eno oceno nižja. Rezultate najboljšega modela predstavimo tudi s histogramom napak konsistentnosti 6.9<sup>1</sup>. Opazimo, da so napake virtualnega sodnika konsistentne z napakami pravih sodnikov.



**Slika 6.8:** Matrika zamenjav, pridobljena z napovedjo ocen obstoječih sodnikov z najboljšim modelom, kjer smo učno množico razširili s tekmo kombinatorcev.

Ekspirimentalno smo ovrednotili tudi način učenja, kjer za učni primer vzamemo oceno vsakega posameznega sodnika posebej in tako pridobimo 5 učnih podatkov za posamezen skok. S tako obliko vhoda, kjer isti vhodni podatek predstavimo večkrat s posameznimi ocenami sodnikov, smo poizku-

<sup>1</sup>Referenčno oceno pridobljeno z robustnim povprečjem smo tukaj zaokrožili na najbližjo polovico.



**Slika 6.9:** Histogram napak konsistentnosti za najboljši model na mali skalnici.

sili uvesti neko vrsto regularizacije. Rezultati se niso statistično pomembno izboljšali. Prav tako ne dosežemo boljših rezultatov z drugim načinom normalizacije, kjer za center vzamemo center detekcije smučarskega skakalca. Vseeno je normalizacije pomembna, saj brez uporabe normalizacije naš model ne konvergira.

Metodo ovrednotimo tudi na prvih serijah tekem na majhni in veliki skalnici. Za učenje uporabimo podatke finalne serije. Rezultati so predstavljeni v tabeli 6.5. Opazimo konsistentnost napovedovanja ocen virtualnega sodnika na ostalih tekmah ter slabše napovedi na prvih serijah tekem. Razlog za slabšo napoved po našem mnenju leži v manjši količini učnih podatkov ter višjem povprečju in manjši razpršenosti učnih podatkov v finalni seriji.

Ker mora biti vhod v arhitekturo konvolucijske nevronske mreže fiksne velikosti, se v vhodu v metodo pojavi implicitni podatek o dolžini skoka, saj je



**Tabela 6.5:** Povprečna absolutna napaka konsistentnosti prve serije tekem ter tekmi na veliki skakalnici.

skakalnica	sodnik 1	sodnik 2	sodnik 3	sodnik 4	sodnik 5	virtualni sodnik
mala (1)	$0.28 \pm 0.20$	$0.15 \pm 0.16$	$0.21 \pm 0.22$	$0.22 \pm 0.22$	$0.16 \pm 0.18$	<b><math>0.36 \pm 0.34</math></b>
velika	$0.17 \pm 0.18$	$0.33 \pm 0.25$	$0.11 \pm 0.14$	$0.18 \pm 0.20$	$0.18 \pm 0.19$	<b><math>0.31 \pm 0.21</math></b>
velika (1)	$0.19 \pm 0.16$	$0.26 \pm 0.21$	$0.18 \pm 0.19$	$0.20 \pm 0.18$	$0.21 \pm 0.21$	<b><math>0.30 \pm 0.31</math></b>

najmanjša velikost vhoda pogojena z najdaljšim skokom. Krajši skoki v delu vhoda nimajo definiranih podatkov in vsebujejo začetno ničelno vrednost. Dolžina skoka uradno ne vpliva na sodniško oceno, v praksi pa kot gledalci zelo pogosto opazimo sodniško pristranskost zaradi dolžine skoka. Da smo preverili, v kolikšni meri dolžina skoka vpliva na sodniško oceno, smo naučili regresijski model, ki napove sodniško oceno iz dolžine skoka. Uporabili smo metodo naključnih gozdov s 50 regresijskimi drevesi in linearno regresijo, ki smo jo naučili za vsakega sodnika posebej.

Povprečna absolutna napaka po posameznem sodniku na mali skakalnici je predstavljena v tabeli 6.6. Napaka napovedi ni dosti boljša od povprečnega sodnika iz prve serije tekme. Opazimo, da z našo metodo dosežemo boljše rezultate, kar pomeni, da razporeditev delov telesa in smuči igra pomembno vlogo pri sodniških ocenah in da naša metoda to izkoristi.

Poravnavo posameznih smučarskih skokov poizkušamo doseči tudi s kubično interpolacijo krajših skokov, tako da so posamezne trajektorije enako dolge in bolje poravnane. Tako pripravljeni podatki ne vsebujejo implicitne informacije o dolžini skoka. Doseženi rezultati so nekoliko slabši, saj dosežemo napako konsistentnosti  **$0.33 \pm 0.27$** . Razlog za slabše rezultate je težko določiti. Poleg izgube implicitne informacije o dolžini skoka smo z interpolacijo umetno spremenili vhodne podatke in morebiti še dodatno poudarili šum v vhodnih podatkih.

Tabela 6.7 povzame najpomembnejše rezultate na mali skakalnici. Ugotovimo lahko, da različni poizkusi izboljšanja predstavitve podatkov niso prinesli bistveno boljših rezultatov. Še najpomembneje je k izboljšanju rezultatov

**Tabela 6.6:** Povprečna absolutna napaka po posameznih sodnikih z uporabo modela, ki napoveduje sodniške ocene obstoječih sodnikov iz dolžine skoka.

Vrsta napovedi	sodnik 1	sodnik 2	sodnik 3	sodnik 4	sodnik 5
naključni gozd	$0.42 \pm 0.32$	$0.51 \pm 0.37$	$0.53 \pm 0.38$	$0.50 \pm 0.48$	$0.48 \pm 0.35$
linearna regresija	$0.53 \pm 0.30$	$0.66 \pm 0.48$	$0.67 \pm 0.43$	$0.56 \pm 0.46$	$0.61 \pm 0.50$

pripomoglo večje število učnih podatkov. Po našem mnenju bi rezultate bistveno izboljšali tudi z namenskimi kamerami, s čimer bi izboljšali zanesljivost detekcij in posledični zmanjšali količino šuma v vhodnih podatkih.

**Tabela 6.7:** Povzetek rezultatov na mali skakalnici.

A <sup>1</sup>	B <sup>2</sup>	C <sup>3</sup>	D <sup>4</sup>
0.27	0.25	<b>0.21</b>	0.33

<sup>1</sup> Osnovni rezultati brez dodatnega procesiranja ali dodatnih podatkov.

<sup>2</sup> Uporaba glajenja z mediano s filtrom velikosti  $n = 3$ .

<sup>3</sup> Najboljši doseženi rezultat z dodatnimi podatki.

<sup>4</sup> Uporaba interpolacije za boljšo poravnavo podatkov.

## Poglavje 7

# Zaključek

V magistrskem delu smo razvili metodo za avtomatsko ocenjevanje smučarskih skokov iz videoposnetkov. Kot glavni vir podatkov uporabimo detektirane dele telesa in smuči skozi let smučarskega skakalca. Ker smo k reševanju problema pristopili kot prvi, je bilo potrebno zgraditi zbirko anotiranih tekem smučarskih skokov, kjer smo anotirali skoraj 1800 slikovnih okvirjev, ki jih uporabimo za učenje izbranih metod.

Celoten sistem smo zgradili z uporabo konvolucijskih nevronske mreže. Najpomembnejši del predstavlja metoda za detekcijo delov telesa CPM [13], ki smo ji dodali podporo za detekcijo delov smuči. Na testni zbirki pokažemo, da smo metodo uspešno naučili na domeni smučarskih skokov z dovolj dobro natančnostjo delovanja, ki omogoča nadaljnjo uporabo detektiranih delov telesa in smuči za napoved sodniških ocen.

Kot se izkaže, so pravila za ocenjevanje sloga smučarskih skokov definirana zelo subjektivno in jih ni mogoče opisati v matematični obliki. Metodo za ocenjevanje sloga zato naučimo iz že obstoječih sodniških ocen. Izkaže se, da se metoda lahko uspešno nauči modela za ocenjevanje zgolj iz razporeditve delov telesa in smuči skozi let in iz obstoječih sodniških ocen. Tako naučen model ima napako napovedi, ki je konkurenčna pravim sodnikom.

Glavno težavo pri implementaciji metod je predstavljalo pomanjkanje anotiranih podatkov. Navkljub velikemu številu anotiranih slikovnih okvir-

jev bi za splošnost uporabe metod potrebovali anotacije iz večih različnih prizorišč. Proces anotiranja je časovno zamuden, zato smo začeli razvijati spletno aplikacijo, ki se bo izvajala na platformi Amazon Mechanical Turk. Ta omogoča vključitev širše množice ljudi, ki za majhno plačilo anotirajo posamezne slikovne okvirje iz drugih skakalnic.

Boljše rezultate bi dosegli tudi, če bi imeli dostop do odbitkov sodniških ocen po posameznih fazah skoka, ki jih sodniki vnašajo v posebno napravo za vsakega tekmovalca posebej. S tako obliko podatkov bi bistveno povečali učno množico, saj bi model naučili po posameznih fazah smučarskega skoka. S takim modelom bi tudi lažje določili fazo skoka, ki je odgovorna za slabo oceno, kar bi bila dodatna dodana vrednost.

Naše delo ni omejeno zgolj na domeno smučarskih skokov, saj je mogoče naše ugotovitve in pristope uporabiti tudi pri drugih športih, kjer razporeditev delov telesa prav tako igra pomembno vlogo.

# Literatura

- [1] FIS, The International Ski Competition Rules (ICR), Book 3, Ski Jumping, [http://www.fis-ski.com/mm/Document/documentlibrary/Skijumping/03/19/96/ICRSkiJumping2013\\_English.pdf](http://www.fis-ski.com/mm/Document/documentlibrary/Skijumping/03/19/96/ICRSkiJumping2013_English.pdf), [Dostop: 10. avgust 2017].
- [2] J. Eskildsen, Goal detector for detection of an object passing a goal plane, wO Patent App. PCT/DK2008/000,360 (Apr. 16 2009). URL <https://www.google.com/patents/W02009046722A1?cl=es>
- [3] Hawk-Eye, Goal Line Technology, <https://www.hawkeyeinnovations.com/products/ball-tracking/goal-line-technology>, [Dostop: 10. avgust 2017].
- [4] Hawk-Eye, Electronic Line Calling, <https://www.hawkeyeinnovations.com/products/ball-tracking/electronic-line-calling>, [Dostop: 10. avgust 2017].
- [5] Swiss Timing, Video distance measurement, [https://www.swisstiming.com/fileadmin/Resources/Data/Datasheets/DOCM\\_SJ\\_Competition-System\\_1215\\_EN.pdf](https://www.swisstiming.com/fileadmin/Resources/Data/Datasheets/DOCM_SJ_Competition-System_1215_EN.pdf), [Dostop: 10. avgust 2017].
- [6] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2D Human Pose Estimation: New Benchmark and State of the Art Analysis, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3686–3693.

- 
- [7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: European Conference on Computer Vision (ECCV), Zürich, 2014.
  - [8] L. Pishchulin, M. Andriluka, P. Gehler, B. Schiele, Poselet Conditioned Pictorial Structures, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.
  - [9] L. Pishchulin, M. Andriluka, P. V. Gehler, B. Schiele, Strong Appearance and Expressive Spatial Models for Human Pose Estimation, in: IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, 2013, pp. 3487–3494.
  - [10] D. Ramanan, Learning to parse images of articulated bodies, in: P. B. Schölkopf, J. C. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems 19, MIT Press, 2007, pp. 1129–1136.
  - [11] P. F. Felzenszwalb, D. P. Huttenlocher, Efficient matching of pictorial structures, in: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662), Vol. 2, 2000, pp. 66–73 vol.2.
  - [12] A. Toshev, C. Szegedy, DeepPose: Human Pose Estimation via Deep Neural Networks, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14, IEEE Computer Society, Washington, DC, USA, 2014, pp. 1653–1660.
  - [13] S.-E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, Convolutional Pose Machines, in: CVPR, 2016.
  - [14] A. Bulat, G. Tzimiropoulos, Human pose estimation via Convolutional Part Heatmap Regression, in: ECCV, 2016.
  - [15] E. Shelhamer, J. Long, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 640–651.

- 
- [16] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, B. Schiele, DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation., IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
  - [17] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, B. Schiele, DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model, in: ECCV, 2016.
  - [18] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, in: CVPR, 2017.
  - [19] D. Tome, C. Russell, L. Agapito, Lifting From the Deep: Convolutional 3D Pose Estimation From a Single Image, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
  - [20] S. Johnson, M. Everingham, Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation, in: Proceedings of the British Machine Vision Conference, 2010.
  - [21] S. Johnson, M. Everingham, Learning effective human pose estimation from inaccurate annotation, in: CVPR, 2011, pp. 1465–1472.
  - [22] B. Sapp, B. Taskar, MODEC: Multimodal Decomposable Models for Human Pose Estimation, in: CVPR, 2013.
  - [23] Y. Feng, Y. Li, J. Luo, Learning effective Gait features using LSTM, in: 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 325–330.
  - [24] D. Liu, M. Ye, X. Li, F. Zhang, L. Lin, Memory-based Gait Recognition, in: Proceedings of the British Machine Vision Conference (BMVC), BMVA Press, 2016, pp. 82.1–82.12.
  - [25] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

- 
- [26] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, et al., Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks., in: AAAI, Vol. 2, 2016, p. 8.
- [27] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1110–1118.
- [28] Y. Du, Y. Fu, L. Wang, Skeleton based action recognition with convolutional neural network, in: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015, pp. 579–583.
- [29] M. A. Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.  
URL <http://neuralnetworksanddeeplearning.com/>
- [30] A. Karpathy, Stanford CS231n Course Notes (2017).  
URL <http://cs231n.github.io/>
- [31] F. Rosenblatt, Principles of neurodynamics. perceptrons and the theory of brain mechanisms, Tech. rep., Cornell Aeronautical Laboratory (1961).
- [32] M. Minsky, S. Papert, Perceptrons: An Introduction to Computational Geometry, MIT Press, 1969.
- [33] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.
- [34] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.



- 
- [35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR, 2009.
  - [36] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
  - [37] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The Pascal Visual Object Classes Challenge: A Retrospective, *International Journal of Computer Vision* 111 (1) (2015) 98–136.
  - [38] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in: *Neural Information Processing Systems (NIPS)*, 2015.
  - [39] J. Uijlings, K. van de Sande, T. Gevers, A. Smeulders, Selective Search for Object Recognition, *International Journal of Computer Vision*.
  - [40] R. Girshick, Fast R-CNN, in: *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.
  - [41] V. Ramakrishna, D. Munoz, M. Hebert, A. J. Bagnell, Y. Sheikh, Pose Machines: Articulated Pose Estimation via Inference Machines, in: *ECCV*, 2014.
  - [42] L. Breiman, Random Forests, *Machine learning* 45 (1) (2001) 5–32.
  - [43] N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, in: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01, CVPR '05*, 2005, pp. 886–893.

- [44] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv preprint arXiv:1408.5093.
- [45] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single Shot MultiBox Detector, in: ECCV, 2016.
- [46] Y. Yang, D. Ramanan, Articulated Human Detection with Flexible Mixtures of Parts, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2013) 2878–2890.